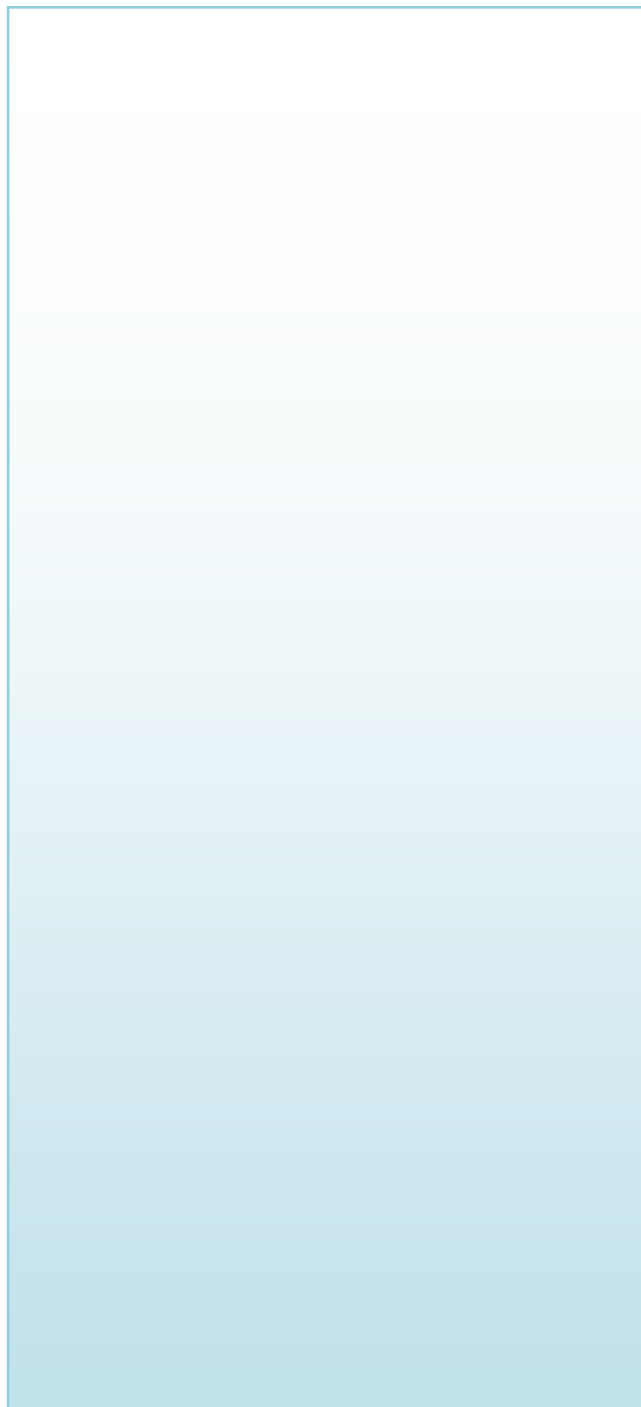


IranDataMiner.ir



بسم الله الرحمن الرحيم

کتاب داده کاوی در R در سال ۲۰۱۱ چاپ شده است. با تعدادی از دوستان بر آن شدیم تا با ترجمه آن را در اختیار قرار علم آموزان قرار دهیم تا نقشی در پیشرفت علمی جامعه در زمینه داده کاوی که یکی از ۱۰ حوزه برتر فناوری اطلاعات است، داشته باشیم.

دلیل استفاده از R کمابز بودن آن است و شاید یکی از مهم ترین دلایل پرکاربرد بودن آن بین داده کاوان است. در سال گذشته کوی سبست رازا کهنیتان را بود و پرکاربردترین ابزار داده کاوی شد.

این کتاب را در اختیار شما عزیزان قرار می دهیم و آن را به امام عصر (عج) تقدیم می کنیم. امید است که روزی بماند قرون ۱۲ و ۱۳ باز هم کشورمان مرجع علمی دنیا شود.

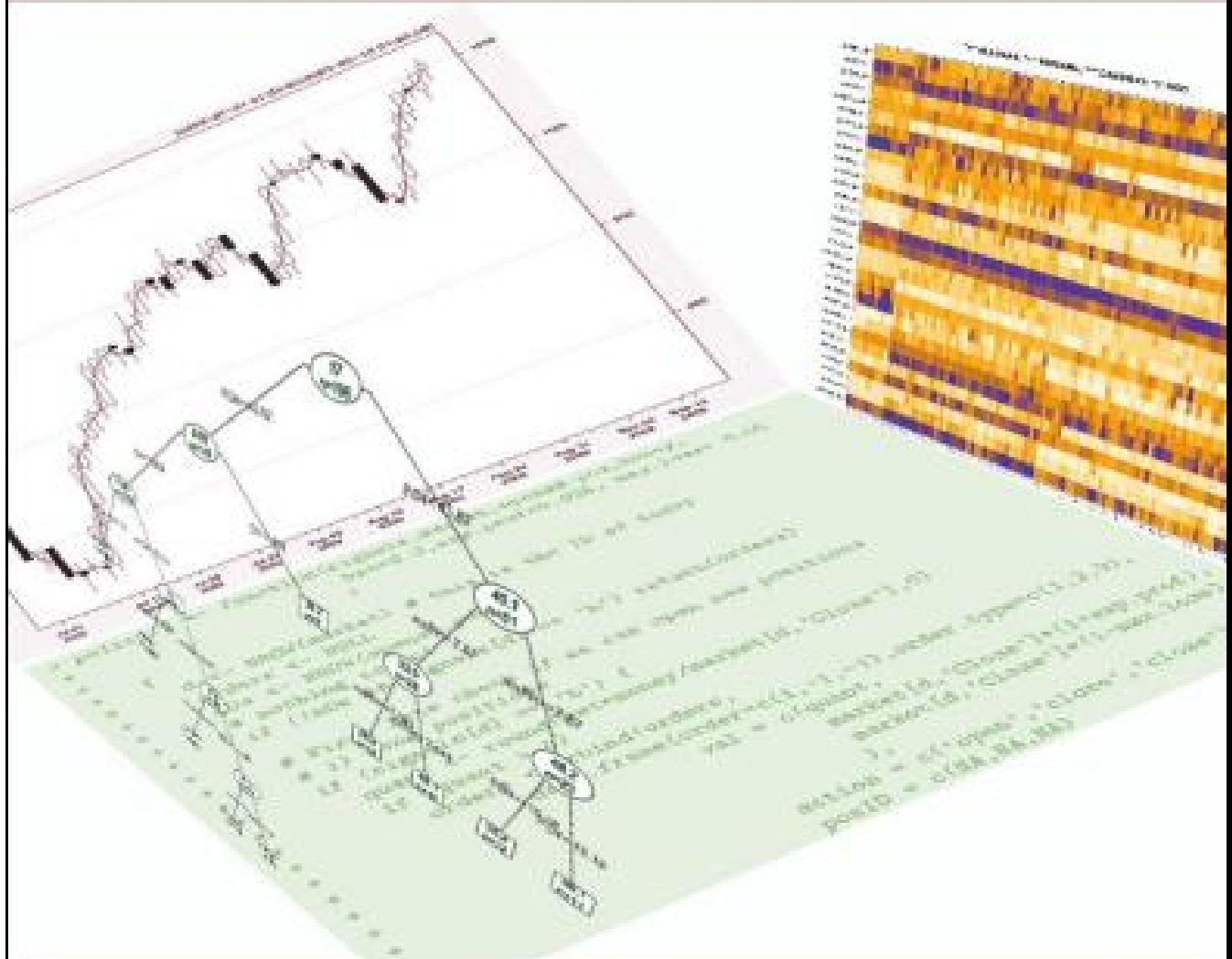
نسخه اصلی کتاب نیز همراه ترجمه فارسی آن نیز برای دانشمندان داده کاوی داده شده است.

مهدی نصیری

Chapman & Hall/CRC
Data Mining and Knowledge Discovery Series

Data Mining with R

Learning with Case Studies



فصل اول

R یک زبان برنامه نویسی و محیطی برای پردازش آماری می باشد. آن شبیه زبان S می باشد که در آزمایشگاه های بل T و AT توسط ریک بکر، جان چمبرز و الان ویکس ساخته شد.

نگارش های R برای خانواده های سیستم عامل یونیکس، ویندوز و مک موجود است. از این گذشته، R بر روی معماری های سیستم کامپیوتری مختلف مانند سیستمهای اینتل، پاور پی سی، آلفا و سیستمهای وابسته اجرا می شود. R در ابتدا توسط ایعماک و جنتلمن (1996)، هر دو از دانشگاه اوکلند نیوزیلند، ساخته شد. ساخت کنونی R توسط یک تیم مرکزی دوازده نفر از مؤسسات مختلف سراسر جهان، انجام شد. ساخت و توسعه R، کمال استفاده را از جامعه رو به رشدی که به خاطر فلسفه منبع باز بودنش، به رشد و توسعه خود کمک می کند، می برد. عملاً، کد منبع هر یک از اجزاء تشکیل دهنده R بطور آزادانه برای بازبینی و یا حتی اقتباس، در دسترس می باشد.

این واقعیت به شما اجازه می دهد تا اعتبار آنچه را که شما در R از آن استفاده می کنید، بررسی و آزمایش کنید. منتقدان زیادی نسبت به مدل منبع باز، وجود دارند. بیشترین آنها، نبود پشتیبان بعنوان یکی از عیب های اساسی نرم افزار منبع باز را عنوان می کنند. این مورد، مسلماً، نقطه ضعفی برای R نیست. تعداد بسیار زیادی از سایتها، کتابها و منابع معتبر وجود دارند که اطلاعات آزادی را از R تهیه می کنند. از این گذشته، فهرست پستی کسانی که از کمک R برای کسب اطلاعات استفاده می کنند، یک منبع اطلاعاتی و مشورتی ارزشمندی است که بسیار بهتر از هر مقدار پولی است که بتوان با آن برای همیشه خرید کرد. همچنین آرشیو فهرست های پستی قابل جستجو موجود است که شما از قبل از ارسال یک سوال یا درخواست می توانید و باید از آنها استفاده نمایید. اطلاعات بیشتر در مورد این فهرست های پستی را می توانید در سایت اینترنتی R در بخش فهرست های پستی بدست آورید.

استخراج داده ها باید به صورت کشف دانش قابل فهم، پیش بینی نشده، معتبر و با ارزش و مفید از اطلاعات و داده ها انجام شود. این اهداف کلی به طور بدیهی توسط نظامات دیگری همچون آمار، آموزش ماشین، یا شناخت الگو، به طور مشترک انجام می شوند. یکی از مهمترین موضوعات تشخیصی در استخراج اطلاعات، اندازه اطلاعات می باشد. با استفاده گسترده از تکنولوژی کامپیوتر و سیستمهای اطلاعاتی مربوط به آن، مقادیر داده های در دسترس برای مطالعه و تحقیق به طور تصاعدی افزایش یافته است. این مسأله، چالش های سختی را در برابر روشهای تجزیه و تحلیل استاندارد داده ها، بوجود می آورد؛ شخص باید مسائلی مانند کارایی کامپیوتری،

امکانات حافظه ای محدود، ارتباط با پایگاه داده ها و غیره را در نظر بگیرد. همه این مسائل، استخراج اطلاعات را به یک موضوع میان رشته ای تبدیل می کند که نه تنها شامل حال وظایف پژوهشگران اطلاعات خاص می گردد بلکه همچنین افرادی را که با پایگاه داده ها، تجسم اطلاعات در ابعاد کلان، و غیره سر و کار دارند، تحت تأثیر قرار می دهد.

R دارای محدودیت های در اداره مجموعه داده های بسیار زیاد می باشد چرا که همه محاسبات در داخل حافظه اصلی کامپیوتر انجام می شوند. این بدان معنی نیست که ما توانایی مدیریت این مسائل و مشکلات را نخواهیم داشت. با بهره گیری کامل از واسطه های بسیار انعطاف پذیر پایگاه داده ها که در مورد مسائل کلان، انجام دهیم. اگر فلسفه منبع باز را باور داشته باشیم، ما می توانیم از سیستم مدیریت عالی پایگاه داده ها به نام MySQL استفاده نماییم. MySQL همچنین برای تقریباً مجموعه بزرگی از سیستم عامل های کامپیوتری و سیستم های عملیاتی در دسترس می باشد. از این گذشته، R دارای برنامه ای (بسته به نرم افزاری) است که امکان ایجاد ارتباط آسان با MySQL را فراهم می کند (بسته نرم افزاری RMySQL (جیمز و دیبروی، 2009).

جان کلام اینکه، ما امیدواریم، که در پایان مطالعه این کتاب شما متقاعد باشید شوید که می توانید بدون اینکه مجبور به پرداخت هرگونه پولی باشید، استخراج اطلاعات را در مورد مسائل کلان، انجام دهید. این مهم تنها به خاطر تلاش و مساعدت ارزشمند و بزرگوارانه تعداد زیادی از افرادی می باشد که چنین ابزار خارق العاده ای همچون R و MySQL را ساخته اند.

1-1- چگونه این کتاب را مطالعه کنیم؟

روح اصلی ماورای جسم این کتاب، یادگیری با انجام دادن می باشد.

این کتاب بعنوان مجموعه ای از مطالعات موردی، سازماندهی شده است. راه حلهای مورد نیاز برای این مطالعات موردی با استفاده از R به دست می آیند. تمام گامهای لازم برای رسیدن به این راه حل ها تشریح شده اند. با استفاده از سایت اینترنتی این کتاب و برنامه نرم افزاری R که بعنوان کمکی همراه این کتاب می باشد، شما قادر خواهید بود کلیه کدهای موجود در این سند را به همراه اطلاعات راجع به مطالعات موردی به دست آورید. این امر باید برای شما در امتحان کردن آنها تمهیداتی فراهم آورد. در واقع شما باید این متن را در کنار کامپیوتر خودتان مطالعه کنید با هر گامی که بر می دارید تلاش کنید همانطوری که برای شما در این کتاب نشان داده شده است.

کد R با استفاده از فونت زیر در این کتاب نشان داده شده است:

فرمانهای R با اعلان فرمان R، به شکل ">" وارد می شوند. هر زمان شما این اعلان را ببینید می توانید این طور قلمداد کنید که این اعلان یعنی R منتظر شماست تا یک فرمان را وارد کنید. شما فرمان ها را در این اعلان بنویسید و سپس دکمه اینتر را بزنید تا درخواست شما بوسیله R، اجرا شود. این کار ممکن است منجر به ایجاد شکلهایی از خروجی شود و یا اصلاً خروجی نداشته باشد (نتیجه فرمان) و سپس یک اعلان جدید ظاهر می شود. در این اعلان شما ممکن است از کلیدهای جهت دار برای جستجو و ویرایش فرمانهایی که قبلاً وارد شده اند، استفاده نمایید. وقتی شما بخواهید فرمانهایی را شبیه به آنچه که قبلاً انجام داده اید، تایپ کنید، برای اجتناب از تایپ مجدد آنها، این کار مفید خواهد بود.

با این همه شما می توانید از کد ارائه شده در سایت اینترنتی این کتاب، کمال استفاده را ببرید و بین آنچه که شما جستجو یا ویرایش می کنید و میز فرمان R، کدها را ببرید و بچسبانید، بنابراین از اجبار به تایپ تمام فرمانهای مشروح در این کتاب، اجتناب کنید. مطمئناً این مسأله، تجربه یادگیری شما را آسان می کند و درک بالقوه شما را از آن بهبود می بخشد.

2-1- مقدمه کوتاهی در باب R

هدف از این بخش، فراهم آوردن مقدمه کوتاهی در مورد مسائل کلیدی زبان R می باشد.

ما هیچگونه آشنایی با برنامه نویسی کامپیوتر را در نظر نمی گیریم. خوانندگان کتاب باید قادر باشند به آسانی، مثالهای ارائه شده در این بخش را دنبال کنند. هنوز اگر شما انگیزه ای برای ادامه مطالعه این اطلاعات مقدماتی ندارید، نگران نباشید. شاید شما به قسمت مطالعات موردی پیشروی کنید و دوباره به این مقدمه برگردید زیرا به وسیله کاربردهای عینی و واقعی، انگیزه بیشتری را بدست خواهید آورد.

R یک زبان عملیاتی برای آثار گرافیکی و نموداری و محاسبات آماری می باشد. آن می تواند بعنوان گویشی از زبان S قلمداد شود (زبان S ساخته شده در T و AT) که جان چمبرز جایزه ساخت افزاری انجمن ماشین آلات کامپیوتری (ACM) را در 1998 به خاطر آن دریافت کرد. شعارش این بود که این زبان «برای همیشه، شیوه تجزیه و تحلیل، تصور و حتی نحوه دستکاری در اطلاعات افراد را تغییر داد».

R تنها با استفاده از آن در یک مدل تعاملی و دوسویه در خط فرمان آن، می تواند نسبتاً مفید واقع شود. هنوز، استفاده های پیشرفته تر این سیستم می تواند کاربر را به توسعه عملیاتش برای سیستماتیک کردن تکالیف تکراری خود رهنمون شود.

یا اینکه حتی برخی کاربردهای برنامه های نرم افزاری اضافه شونده موجود را با بهره گیری کامل از منابع باز موجود، افزایش داده یا تغییر دهند.

1-2-1- شروع به کار با R

به منظور نصب R در سیستمتان، آسان ترین راه به دست آوردن یک توزیع دودویی از سایت اینترنتی R می باشد جایی که شما می توانید حلقه ای را که شما را به سایت CRAN (شبکه آرشیو گسترده و کامل R) متصل می کند دنبال کنید تا از میان چیزهای دیگر، توزیع دودویی را برای معماری سیستمی یا سیستم عامل ویژه تان بیابید. چنانچه شما ترجیحاً خواستار ساخت R مستقیماً از طریق این منابع باشید، به راحتی می توانید دستورالعملهای چگونگی انجام آن را از CRAN به دست آورید.

بعد از دانلود توزیع دودویی برای سیستم عامل خود، شما فقط نیاز است که دستورالعمل های همراه آن را دنبال کنید. در مورد نگارش ویندوز، شما به راحتی، فایل دانلود شده را اجرا کنید (R-2-10-1-win32-exe) و گزینه هایی را که می خواهید در منوها دنبال کنید، انتخاب نمایید. در برخی از سیستم عامل ها لازم است که با مدیر سیستم تان تماس بگیرید تا به خاطر نبود اجازه برای نصب نرم افزار، عمل نصب برای شما محقق شود.

برای اجرای R در ویندوز شما به سادگی، آیکون مربوط را در صفحه دسکتاپ دابل کلیک می کنید، ولی در نگارش های یونیکس شما باید R را با اعلان سیستم عامل خود تایپ کنید. هردو، کنسول R را با اعلان آن ">" عنوان می کنند.

اگر بخواهید از R خارج شوید، می توانید فرمان (q) را در اعلان صادر کنید. اگر خواستار ذخیره کردن محل کار فعلی تان باشید از شما پرسیده می شود.

اگر شما فقط بخواهید تجزیه و تحلیل فعلی تان را در نقطه ای که قطع می کنید، بعداً از نو شروع کرده و ادامه بدهید، باید به این سؤال جواب بله بدهید.

هرچند مجموعه ابزارهایی که به همراه R ارائه می شوند به تنهایی تقریباً قدرتمند می باشند، با این حال طبیعی است اگر کار شما به جایی برسد که بخواهید تعدادی از برنامه های بزرگ و (در حال رشد) اضافه شوند موجود در CRAN را که در مورد R می باشد نصب کنید. در نگارش ویندوز این کار به آسانی از طریق منوی «برنامه های نرم افزاری» قابل انجام است. بعد از وصل کردن کامپیوترتان به اینترنت شما باید گزینه «برنامه نرم افزاری نصب از CRAN ..» را از این منو انتخاب کنید.

این گزینه فهرستی از برنامه های نرم افزاری موجود در CRAN را نشان خواهد داد.

شما آنهایی را که یا یکی از آنهایی را که می خواهید انتخاب کنید، R آن برنامه های نرم افزاری یا یکی از آنها را آن طور که شما خواسته اید، دانلود کرده و بطور خودکار، آن یا آنها را بر روی سیستم شما نصب خواهد کرد. در نگارش های یونیکس، چیزها ممکن است قدری نسبت به امکانات گرافیکی نصب R شما متفاوت باشند. هنوز حتی بدون انتخاب از منوها، این عمل ساده است. خوب است که این برنامه نرم افزاری را که توبعی را برای ایجاد ارتباط با پایگاه داده های MySQL فراهم می کند، دانلود کنید. نام این برنامه، RMySQL می باشد. شما فقط کافی است که فرمان زیر را در اعلان R تایپ کنید:

نوسان در خط

تابع مذکور دارای پارامترهای فراوانی است که در میان آنها شناسه منبع وجود دارد که به شما اجازه می دهد تا نزدیکترین انعکاس CRAN را نشان دهید. با این همه، اولین باری که شما این تابع را در یک جلسه R اجرا می کنید، آن به شما منبع اطلاعاتی را که می خواهید از آن استفاده کنید، اعلام می کند.

یک کار که شما مطمئناً باید انجام دهید، نصب برنامه نرم افزاری مرتبط با این کتاب است. که امکان دستیابی شما را به چند تابع استفاده شده در سراسر این کتاب و پایگاه داده ها فراهم می کند. برای نصب آن، به همان ترتیبی که در مورد سایر برنامه های نرم افزاری، رایج است عمل کنید:

اگر شما می خواهید از برنامه نرم افزار اخیراً نصب شده در کامپیوترتان، اطلاع کسب نمایید شما می توانید فرمان زیر را صادر کنید: (>installed.packages)

این کار یک خروجی طولانی در پی دارد که هر خط آن دارای برنامه نرم افزاری، اطلاعات نگارش آن، برنامه های نرم افزار که این برنامه بر پایه آنها تهیه شده است و غیره می باشد. فهرست برنامه های نرم افزاری کار بر پسند که در عین حال از کامل بودن کمتری برخوردار است با صادر کردن فرمان زیر می تواند بدست آید:

>library ()

فرمان زیر می تواند بسیار مفید باشد چون آن به شما امکان می دهد تا بررسی کنید چه نگارشهای جدیدتری نسبت به نگارش برنامه های نرم افزاری نصب شده شما، در CRAN موجود است:

>old.packages ()

از این گذشته، شما می توانید برای به روز رسانی تمام برنامه های نرم افزاری نصب شده تان از فرمان زیر استفاده کنید:

>Update. Packages ()

R دارای یک سیستم یکپارچه کمک رسانی می باشد که شما می توانید برای کسب اطلاعات بیشتر در باره سیستمتان و کاربردهای آن، از آن استفاده نمایید. علاوه بر این، شما می توانید ارائه سند و مدرک اضافی را در سایت R بیابید، R به همراه مجموعه ای از فایل های HTML ارائه می شود که با استفاده از یک مرورگر وب می تواند مورد دسترس می باشند. در نگارش ویندوز مربوط به R، این صفحات از طریق منوی HELP در دسترس می باشند. راه دیگر این است که شما می توانید فرمان () help.start را در این اعلان صادر کنید تا یک مرورگر شروع به نشان دادن صفحات کمک رسانی HTML نماید. شکل دیگر بدست آوردن صفحات کمک رسانی، استفاده از تابع () help می باشد. بعنوان مثال، اگر شما در مورد تابع () plot، نیاز به کمک داشته باشید، شما می توانید فرمان Help (plot) (یا به جای آن، فرمان ؟plot) را وارد نمایید. یک گزینه نسبتاً قدرتمند، به شرط اینکه شما به اینترنت متصل باشید، استفاده از تابع () R Site Search می باشد که یا کلمات کلیدی را جستجو می کند یا اینکه صفحات کمک رسانی، دفترچه های راهنمای R و آرشیو فهرست های پستی را ارائه می کند؛ بعنوان مثال،

>RSiteSearch('neural network')

نهایتاً اینکه چندجایی در وب وجود دارند که کمک هایی را در مورد جنبه هایی از R، از قبیل <http://www.rseek.org/> ارائه می دهند.

2-2-2- اشياء R

دو مفهوم اصلی در ورای زبان R وجود دارند: اشياء و توابع.

یک شیء می تواند بعنوان یک فضای ذخیره سازی با یک نام وابسته و مربوط، در نظر گرفته شود. هر چیزی که در R وجود دارد در داخل یک شیء ذخیره می شود. تمام متغیرها، داده ها، توابع و غیره.. در حافظه کامپیوتر به شکلی ذخیره می شوند که این شکل یا اشکال، اشیاء نامیده می شوند.

توابع، نوع ویژه ای از اشیاء R هستند که برای انجام برخی عملیات، طراحی شده اند. آنها معمولاً برخی از شناسه ها را در نظر می گیرند و خروجی یا نتیجه ای را بوسیله اجرای برخی از مجموعه عملیات ها (خودشان معمولاً تابع دیگری را اجرا می کنند) به دست می دهند.

R قبلاً به همراه مجموعه بسیار قدرتمندی از توابع موجود به منظور استفاده ما، ارائه می شد، اما همانطوری که ما بعداً خواهیم دید، کاربر هم می تواند توابع جدید ایجاد نماید.

محتوا ممکن است در اشیاء با استفاده از این کاربر جایگزین ذخیره شود. این کاربر می تواند به صورت یک پرانتز زاویه دار که به دنبال آن یک علامت منها قرار دارد، نشان داده شود. ($<-$):

$x <- 945$

در نتیجه

دستورالعمل قبلی باید به این ترتیب، شماره 945 را برای یک شیء به نام x ذخیره کند.

فرد با وارد کردن ساده نام یک شیء در اعلان R می تواند محتویات آن را مشاهده کند:

x

945 [1]

علامت "[1]" که تقریباً به صورت یک رمز در جلوی شماره 945 قرار گرفته است می تواند بدین صورت خوانده شود این خط نشانگر مقادیری است که از اولین عنصر یک شیء شروع می شود. این علامت مخصوصاً برای اشیایی که دارای چندین مقدار می باشند، همانند بردارها که بعداً خواهیم دید، مؤثر و مفید می باشند.

در پایین، شما مثالهای دیگری از دستورهای جایگزینی را خواهید یافت. این مثالها باید این مسأله را روشن نمایند که این یک عملکرد مخرب است که هر شیء در هر زمان t نقاط دارای یک محتوای تنها می تواند باشد. این بدان معنی است که با جایگزین کردن بعضی از محتواها جدید با یک شیء موجود، شما در واقع، محتوای قبلی آن را از دست می دهید:

>y<-39

>y<-43

>y

>y

[1]39

[1] 43

شما همچنین می توانید عبارات عددی را با یک شیء، جایگزین نمایید. در این مورد، شیء نتیجه عبارت را ذخیره خواهد کرد:

>z<-5

>i<-(z*2+45)/2

>w<-z^2

>i

[1]25

[1]27.5

این بدان معنی است که ما می توانیم این عمل جایگزینی را مفهومی بدانیم به این عبارت که محاسبه نمایید آنچه را که در سمت راست اپراتور آمده است، و جایگزین (ذخیره) کنید نتیجه این محاسبه را با شیء که نام آن در سمت چپ مشخص شده است.

اگر شما فقط بخواهید نتیجه برخی از عملیات حسابی را بدانید، شما نیازی به جایگزین کردن نتیجه یک عبارت یا یک شیء را ندارید. عملاً شما می توانید اعلان R را بعنوان نوعی ماشین حساب بکار ببرید:

>(34+90)/12.5

[1] 9/92

هر شیء که شما ایجاد می کنید تا زمانی که آنرا حذف نکرده اید در حافظه کامپیوتر باقی خواهد ماند. شما ممکن است اخیراً اشیاء را در حافظه با صدور فرمان () 1S یا فرمان () objects در اعلان، فهرست کرده باشید. اگر شما نیازی به یک شیء ندارید، می توانید مقداری از فضای حافظه را با حذف آن شیء، آزاد کنید.

>1S()

[1] "i" "w" "X" "y" "z"

>rm(y)

>rm(z,w,i)

نام اشیاء ممکن است هر حرف کوچک یا بزرگ، ارقام تا 9 (بجز در ابتدای نام)، و همچنین نقطه، "0"، که مثل یک حرف عمل می کند، همه اینها را در بر بگیرد. توجه کنید که نامها در R، case sensitive، باشند به این معنی که color با c کوچک دو شیئی متمایز می باشند. در واقع این دلیل مضاعفی است برای ناکامی شروع کنندگانی که با خطاهای شیئی یافت نمی شود مواجه می شوند. اگر شما با این نوع خطا مواجه شوید، درستی نام آن شیئی را که موجب خطا شده است، را بررسی نمایید.

3-2-1- اساسی ترین شیئی داده ای در R، یک بردار است. حتی وقتی که شما یک شماره تنها را با یک شیئی جایگزین می کنید (مثل $x < -45.3$)، شما برداری را که در بردارنده یک عنصر تنها می باشد، خلق می کنید. تمام اشیاء دارای یک مد (وضعیت) و یک طول می باشند. مد بیانگر نوع اطلاعات ذخیره شده در داخل شیئی می باشد. بردارها به منظور ذخیره مجموعه ای از عناصر یک نوع داده هسته ای (تقسیم ناپذیر) بکار می روند. انواع هسته ای مهم عبارتند از کاراکتر، منطقی، عددی و پیچیده. بنابراین شما ممکن است دارای بردارهای کاراکترها، مقادیر منطقی (T یا F یا FALSE یا TRME)، اعداد و اعداد پیچیده باشید. طول یک شیئی عبارت است از شماره (عدد) عنصر در آن، و با تابع length () می تواند به دست آید.

در بیشتر موارد شما از بردارهایی با طول بیشتر از 1 استفاده خواهید کرد. شما می توانید با استفاده از تابع C ()، که شناسه های آن را به شکل یک بردار ترکیب می کند، یک بردار در R بسازید و ایجاد نمایید.

همه عناصر بردار باید به مد یکسان تعلق داشته باشند. اگر بدین صورت نباشد، R آن را وادار به تبدیل نوع خواهد کرد. مثالی در این باب در پایین آمده است:

همه عناصر بردار باید به مد کاراکتر تبدیل شوند. مقادیر کاراکتر، عبارت از رشته های کاراکتری است که بوسیله یک گیومه تنها یا گیومه دوتایی، محصور می شوند. همه بردارها ممکن است در بردارنده مقدار ویژه ای به نام NA باشند.

این نشانگر یک مقدار جا افتاده (گم) می باشد:

شما می توانید از طریق یک شاخص یا نمایه بین گروه ها، به عنصر ویژه یک بردار دسترسی پیدا کنید:

مثال بالا، عنصر دوم بردار Y را معین می کند. شما در بخش 7-2-1 یاد می گیرید که ما از بردارهای شاخص برای دست آوردن شکل های شاخص بندی قوی تر استفاده می کنیم.

شما می توانید مقدار یک عنصر بردار ویژه را با استفاده از استراتژیهای شاخص بندی یکسان بدست آورید:

تغییر طول یک بردار می تواند بوسیله افزودن ساده عناصر بیشتر به آن، با استفاده از یک شاخص که قبلاً وجود خارجی نداشته، عملی شود. بعنوان مثال، بعد از ایجاد بردار خالی X شما می توانید تایپ کنید.

توجه داشته باشید که چگونه اولین دو عنصر دارای یک مقدار ناشناخته، NA می باشند.

این نوع از قابلیت انعطاف پذیری، دارای هزینه خواهد بود. برخلاف زبانهای برنامه نویسی دیگر، در R اگر شما از موقعیت یک بردار که وجود خارجی ندارد، استفاده کنید، با خطا مواجه نخواهید شد:

برای کوچک شدن اندازه یک بردار، شما می توانید کمال استفاده از این واقعی ببرید که همانطوری که قبلاً هم بدان اشاره شد، کاربر جایگزین، مخرب است. بعنوان مثال،

در طی استفاده از شکل های شاخص بندی قوی تر که در قسمت 1-2-7 بررسی خواهد شد، شما قادر خواهید بود عناصر ویژه یک بردار را با روش آسان تری حذف نمایید.

1-2-4- بردارسازی

یکی از قوی ترین ابعاد زبان R ، ساخت بردار از چند تابع موجود در آن می باشد. این توابع مستقیماً بر روی هر عنصر یک بردار عمل می کنند. برای مثال

تابع $\text{sqr}()$ ریشه دوم (جذر) شناسه اش را محاسبه می کند. در این مورد ما از یک بردار اعداد بعنوان شناسه آن استفاده کرده ایم. بردارسازی، این تابع را به سوی ایجاد یک بردار با همان طول، با هر عنصر ناشی از به کار بردن تابع مربوط به بردار اصلی، هدایت می کند. شما همچنین این کارکرد R را برای انجام دادن عملیات حسابی بردار بکار ببرید:

چه می شد اگر بردارها طول یکسان نداشتند؟ R از یک قاعده بازیافت بوسیله تکرار بردار کوتاهتر تا زمانی که آن جایگزین بردار با اندازه بزرگتر شود، استفاده می کند.

این کار به گونه ای است که انگار بردار $C(10,2)$ ، قبلاً بردار $C(10,2,10,2)$ بوده باشد. اگر طول این بردارها، چندتایی نباشند، آنگاه یک هشدار صادر می شود:

طول شی کوتاهتر یک چندتایی با طول شی کوتاهتر نمی باشد.

با این همه، قاعده بازیافت، استفاده شده است و این عمل انجام شده است (آن یک هشدار است نه یک خطا).

همانطوری که ذکر شد، اعداد تنها به عنوان بردارهای با طول 1 در R نشان داده شده اند، این خیلی سودمند است برای عملیاتی مانند این مورد که در پایین نشان داده شده است.

توجه کنید که چگونه عدد 2 (در واقع بردار $C(2)$ در اثر چندگانگی عناصر VI بوسیله 2 بازیافت شده است. همانطوری که ما بعداً خواهیم دید، این قاعده بازیافت، با دیگر اشیاء نیز بکار می رود، اشیایی از قبیل ماتریس ها و آرایه ها.

5-2-1- ضرایب

ضرایب، شکل ساده و فشرده ای از مدیریت داده های طبقه بندی شده (اسمی) را ارائه می دهند. ضرایب، دارای سطوحی هستند که مقادیر ممکن را که آنها می توانند داشته باشند، نشان می دهند. ضرایب، سودمند هستند به ویژه در مجموعه داده ها جایی که شما متغیرهای اسمی با یک عدد ثابت دارای مقادیر احتمالی، دارید. چند تابع گرافیکی و خلاصه را که ما در فصلهای آینده بررسی خواهیم کرد، استفاده کامل را از این اطلاعات خواهند برد. ضرایب به شما اجازه می دهند تا مقادیر متغیرهای اسمی تان را همانطور که هستند، استفاده کرده و نشان دهید، که به وضوح دارای قابلیت تفسیری بیشتری برای کاربر می باشند، حال آنکه R به طور خاصی این مقادیر را بعنوان کدهای عددی که به طور قابل توجهی دارای قابلیت حافظه ای کاراتری می باشند، ذخیره می کند.

بنابراین چگونه ضرایب را در R ایجاد می کنند. بهتر است شما برداری با جنس 10 فرد داشته باشید:

شما می توانید این بردار را به برداری بوسیله وارد کردن، تبدیل نمایید.

توجه داشته باشید که شما دیگر دارای یک بردار کاراکتر نیستید. در واقع، همانطوری که در بالا ذکر شد، ضرایب به طور خاص بعنوان بردارهای عددی نشان داده می شوند. در این مثال، ما دو سطح داریم، f و m که بطور خاص و به ترتیب 1 و 2 نشان داده می شوند. با این وجود، شما نیازی نیست که نگران این باشید، همانطوری که شما می توانید مقادیر کاراکتر اصلی را استفاده نمایید، و R هم زمانی که ضرایب را به شما نشان می دهد، از آنها استفاده خواهد کرد. بنابراین انگیزه ترجمه و تفسیر کدنویسی، که به خاطر دلایل کارایی ایجاد شده است، کاملاً برای شما روشن است.

فرض کنید که شما 5 فرد اضافی داشته باشید که اطلاعات جنس آنها را تصمیم دارید در شیء ضریب دیگر ذخیره کنید. فرض کنید که آنها همه مرد هستند. اگر شما هنوز بخواهید که این شیء ضریب، دو سطح یکسان مثل شیء g را داشته باشد، شما باید از فرمول زیر استفاده کنید:

بدون شناسه سطوح؛ ضریب $other.g$ یک سطح تنهای (m) را خواهد داشت.

بعنوان یک توضیح حاشیه ای، این یکی از ابتدایی ترین مثالهایی است که در مورد یکی از رایج ترین چیزها در یک زبان برنامه نویسی کاربردی مثل R ، که ترکیب تابع می باشد، ارائه شده است. در واقع ما یک تابع ($factor$) را برای نتیجه گرفتن از تابع (c) برای یک شیء بکار می بریم. به طور بدیهی، ما در ابتدا نتیجه تابع (C) را باری یک شیء جایگزین نموده ایم و سپس ضریب این تابع را با این شیء نامگذاری کرده ایم. به هر حال این خیلی زیاد طولانی شده است و در واقع مقداری از حافظه را با ایجاد یک شیء اضافی اشغال می کند، و بنابراین فرد گرایش پیدا می کند که ترکیب تابع را تقریباً به طور تکراری استفاده نماید، اگرچه ما موجب این خطر می شویم که کد ما برای افرادی که آشنایی با این نکته هم راجع به ترکیب تابع ندارند، از لحاظ مطالعه، وضعیت مشکل تری را پیدا کند و براحتی قابل مطالعه و بررسی نباشد.

یکی از چیزهای زیادی که شما با این ضرایب می توانید انجام دهید، محاسبه وقوع هر مقدار احتمالی می باشد. این را امتحان کنید.

تابع ($table$) همچنین می تواند برای به دست آوردن جدول بندی متقاطع چند ضریب بکار رود. فرض کنید ما به بردار دیگری، گروه سنی 10 فرد ذخیره شده در بردار g را بدهیم. شما میتوانید به ایجاد جدول متقاطع برای این دو بردار پردازید همانند زیر:

یک توضیح حاشیه ای کوتاه:

شما شاید توجه کرده باشید که بعضی وقتها ما خطی داریم که با علامت «+» شروع شده است. این مساله زمانی اتفاق می افتد که یک خط به اندازه بسیار زیادی در حال بزرگ شدن است و شما تصمیم دارید تا آن را به یک خط جدید تبدیل کنید (بوسیله زدن دکمه اینتر) قبل از آنکه فرمانی را که شما در حال وارد کردن آن می باشید به پایان برسد. چون این فرمان ناقص است و کامل نمی باشد، R خط جدیدی را با اعلان ادامه دار با علامت «+» شروع میکند شما باید به خاطر داشته باشید که این علامتها نباید توسط شما وارد شوند! آنها به طور خودکار توسط R چاپ می شوند (همانند اعلان طبیعی ">").

بعضی اوقات ما می خواهیم تا تکرارهای مرتبط نهایی با این نوع از جداول احتمالی را محاسبه نماییم . داده های پایین ، جمع دو ضریب جنسی و سنی این مجموعه از داده ها را ارائه کرده است :

"1" و "2" در این توابع نشاندهنده بعد اول و دوم این جدول است که ردیف ها و ستون های جدول می باشند.

در مورد تکرارهای مربوط در خصوص هر حاشیه و به طور کلی ما انجام می دهیم :

توجه داشته باشید که اگر ما به جای آن ، درصد آن را بخواهیم ، ما می توانیم به سادگی فراخوان های این تابع را در 100 ضرب نماییم.

1.2.6 توالی های مولد

R دارای چند نمونه امکانات برای تولید انواع مختلفی از توالی ها می باشد . بعنوان مثال اگر شما بخواهید که یک بردار حاوی اعداد صحیح بین 1 تا 1000 را ایجاد کنید شما می توانید به سادگی تایپ زیر را انجام دهید :

که یک بردار به نام X که دارای 1000 عنصر میباشد ایجاد می کند- اعداد صحیح از 1 تا 1000 .

شما باید نسبت به اولویت کاربر " : " دقت داشته باشید . مثالهای زیر این خطر را نشان میدهند:

لطفا مطمئن شوید که آنچه را که در فرمان اول رخ داد متوجه شده اید (قاعده بازیافت را به یاد بیاورید !)

شما شاید توالی های کاهشی مانند مثال زیر ، ایجاد نمایید :

برای تولید توالی های اعداد حقیقی ، می توانید از تابع (seq) استفاده نمایید :

این دستور العمل یک توالی از اعداد حقیقی را بین -4 و 1 با افزایش 0.5 ایجاد نماید . تعداد دیگری از مثالهای راجع به استفاده از تابع (seq) در پایین ارائه داده شده است :

شاید شما توجه کرده باشید که در مثالهای بالا ، شناسه های بکار رفته در فراخوان های تابع ، تا حدی متفاوت از اولین باری که نام پارامتر نشان داده شده ، مشخص شده اند و متفاوت از مقداری که ما می خواهیم برای آن پارامتر مشخص شده استفاده نماییم .

این خیلی سودمند است زمانیکه ما توابع را به همراه تعداد زیادی پارامتر داشته باشیم که دارای بیشترین مقدار پیش فرض باشند . این پیش فرض ها اجازه می دهند به ما تا از اینکه اجبارا آنها را در فراخوانهایمان مشخص

کنیم اجتناب نماییم اگر که این مقادیر با نیازهای ما مطابقت داشته باشند . به هر حال اگر تعدادی از این پیش فرض ها را برای مساله خود بکار نبریم ، لازم است تا مقادیر جایگزین را فراهم کنیم . بدون وارد کردن مشخصات اسمی که در مثالهای بالا نشان داده شده است ، ما نیاز خواهیم داشت تا بسته به موقعیت از این مشخصات استفاده نماییم . اگر پیش فرض پارامتری را که ما می خواهیم تغییر دهیم از پارامترهای آخر این تابع می باشد ، فراخوان بسته به موقعیت ، نیاز به مشخصات مقادیر پارامترهای قبلی دارد ، گر چه ما خواسته باشیم تا از مقادیر پیش فرض آنها استفاده نماییم . با کمک مشخصات اسمی ما از این مشکل دور خواهیم بود همانطوریکه این مورد ، این امکان را به ما می دهد تا به ترتیب پارامترها را در فراخوان های تابعمان تغییر دهیم ، همانطور هم آنها طبق اسمشان مشخص می شوند .

تابع خیلی سودمند دیگری که توالی ها را با الگوی مشخصی تولید می کند عبارت است از تابع (rep :

تابع (g1 می تواند برای تولید توالی های دربر گیرنده ضرایب بکار رود . ساختار دستوری این تابع بصورت (n و k) g1 می باشد که k تعداد سطوح ضریب ، و n تعداد تکرارهای هر سطح می باشد . دو مثال در این زمینه به شرح زیر است :

در نهایت ، R دارای چند تابع می باشد که برای تولید توالی های تصادفی طبق توابع با تراکم احتمالی مختلف ، بکار می رود . توابعی که دارای ساختار مولد (... و par2 و par1) func r می باشند ، جایی که func ، نام توزیع احتمالات است ، n تعداد داده ها برای تولید کردن و par1 و par2 و ... مقادیر بعضی از پارامترهای تابع تراکم می باشند که ممکن است مورد نیاز واقع شوند . بعنوان مثال اگر شما 10 عدد تصادفا تولید شده را از یک توزیع نرمال با میانگین صفر و انحراف از استاندارد واحد لازم داشته باشید ، تایپ کنید :

برای بدست آوردن 5 شماره به طور تصادفی طراحی شده از توزیع t یک دانش آموز با 10 درجه آزادی ، تایپ کنید :

R دارای توابع احتمالات بسیار زیادی است و دارای توابع دیگری که برای بدست آوردن تراکم های احتمالی ، تراکم های احتمالی فزاینده ، و مقادیر این توزیع ها بکار می روند .

1.2.7 . ما قبلا مثالهایی را در مورد اینکه چگونه یک عنصر را از یک بردار بدست بیاوریم در حالی که وضعیت آن را در داخل کروشه نشان دهیم . R همچنین این امکان را به شما می دهد تا از بردارها در داخل پرانتزها

استفاده نمایید . چند نوع بردار شاخص وجود دارند . بردارهای شاخص منطقی ، این عناصر را مطابق با مقادیر حقیقی استخراج می کنند . به یک مثال عینی توجه کنید :

ساختار دوم کد نشان داده شده در بالا یک شرط منطقی است . همانطوریکه X یک بردار می باشد ، این مقایسه برای همه عناصر بردار انجام می شود (قاعده مشهور بازیافت را به خاطر آورید) ، بنابر این برای تولید یک بردار با بسیاری از مقادیر منطقی ، عناصری در X وجود دارند . اگر ما از این بردار با مقادیر منطقی برای شاخص X استفاده کنیم ، ما نتیجه ای از موقعیت X که مطابق با مقادیر حقیقی می باشد ، بدست می آوریم :

که این به این صورت خوانده می شود : وضعیت های X را برای هر یک از عبارتهای منطقی زیر که حقیقی است به من بدهید . توجه داشته باشید که این مورد ، مثال دیگری برای تصور ترکیب تابع می باشد که ما نسبتا به کرات از آن استفاده می کنیم . با بهره گیری کامل از کاربرهای منطقی موجود در R ، شما می توانید از بردارهای شاخص منطقی پیچیده تری استفاده نمایید ، مثل مثال زیر :

شاید همانطوریکه شما حدس زده باشید ، کاربر "1" گسست منطقی را ایجاد می کند ، در حالیکه کاربر "&" برای ایجاد ارتباط منطقی بکار می رود . این بدان معنی است که دستور العمل اول به ما نشان می دهد که عناصر X کمتر یا مساوی 2- یا بزرگتر از 5 می باشند . مثال دوم نشان می دهد که عناصر X ، هر دوی آنها بزرگتر از 40 و کوچکتر از 100 می باشند .

R همچنین این امکان را به شما می دهد که از یک بردار اعداد صحیح برای استخراج چند عنصر از یک بردار استفاده نمایید . اعداد داخل بردار شاخص ها نشان دهنده وضعیتهایی در بردار اصلی است که باید استخراج شوند :

راه دیگر اینکه ، شما از یک بردار با شاخص های منفی می توانید برای نشان دادن عناصری که باید از انتخاب آنها جلوگیری شود ، استفاده نمایید :

به موضوع نیاز به پرانتزها در مثال قبلی به خاطر اولویت کاربر " : " دقت کنید . شاخص ها همچنین می توانند بوسیله یک بردار شامل رشته ها ، کمال استفاده را از این حقیقت ببرند که R این امکان را به شما می دهد که عناصر یک بردار را بوسیله تابع () `names` نامگذاری کنید . عناصر نامگذاری شده در بعضی از مواقع بعلت اینکه وضعیت آنها برای ذخیره کردن آسانتر است ، ارجحیت دارند . بعنوان مثال ، تصور کنید که شما برداری

دارای مقادیر یک پارامتر شیمیایی دارید که در پنج جای مختلف بدست آمده است . شما می توانید که یک بردار نامگذاری شده به شرح زیر ایجاد نکمایید :

در واقع ، اگر شما قبلا از نام این وضعیت ها در این بردار در زمان ایجاد آ » ، اطلاع داشته باشید ، آسانتر است که انجام دهید از این طریق :

بردار PH هم اکنون می تواند با استفاده از نامهای نشان داده شده بالا ، شاخص گذاری شود :

دست آخر اینکه ، شاخص ها شاید تهی باشند ، به این معنی که تمام عناصر انتخاب شده باشند . یک شاخص تهی نشان دهنده این است که در مرحله انتخاب ، محدودیتی وجود ندارد . برای مثال ، اگر شما بخواهید یک بردار را با صفرها پر کنید ، شما به سادگی می توانید انجام دهید " $X[-0]$ " . لطفا توجه داشته باشید که این مورد ، از انجام " $X < -0$ " متفاوت است . این مورد اخیر ، یک بردار با عنصر تنها (صفر) را برای X معین می کند ، در حالی که مورد پیشین (با فرض اینکه X از قبل وجود داشته باشد ، البته !) تمام عنصرهای جاری X را با صفرها پر می کند . هر دو را امتحان کنید !

1.2.8 . ماتریس ها و آرایه ها

عناصر داده ای می توانند در یک شیئی با بیش از یک بعد ذخیره شوند . این مسئله در چند موقعیت ممکن است سودمند واقع گردد . آرایه ها عناصر داده ای را در چند بعد ذخیره مکی کنند . ماتریس ها نوع خاصی از آرایه ها هستند که دارای دو بعد تنها می باشند . آرایه ها و ماتریس ها در R چیزی بیشتر از بردارها با یک علامت خاص به نام بعد ، نمی باشند . به یک مثال توجه کنید . فرض کنید شما دارای بردار اعداد (23 و 78 و 12 و 56 و 44 و 33 و 77 و 66 و 23 و 45) C می باشید . حالت زیر این ده عدد را بعنوان یک ماتریس ، سازماندهی کرده است :

توجه داشته باشید که چگونه اعداد از طریق یک ماتریس با دو ردیف و پنج ستون ، پخش می شوند . (این بعد را ما برای m با استفاده از تابع (\dim مشخص کرده ایم) .

در حقیقت شما می توانید این ماتریس را با استفاده از این دستور العمل ساده تر ، به سادگی ایجاد نمایید :

شاید شما توجه کرده باشید که این بردار اعداد بوسیله ستونها در ماتریس پخش شده است ؛ بدین ترتیب که اولین عدد در اولین ستون جای گرفته است ، سپس دومیین عدد و الی آخر . شما می توانید این ماتریس را با ردیف ها با استفاده از پارامتر تابع `Matrix()` که در پایین آمده است ، پر نمایید :

همانطور که نمایش ماتریس ها نشان می دهد ، شما می توانید عناصر یک ماتریس را از طریق یک طرح شاخص مشابه ، همانند موردی که در بردارها وجود دارد ، به دست آورید ، اما این بار با دو شاخص (ابعاد یک ماتریس) :

شما می توانید با بهره گیری کامل از طرحهای زیر مجموعه ای که در بخش 1.2.7 شرح داده شده است استخراج عناصر یک ماتریس را انجام دهید ، همانطوریکه در مثال زیر نشان داده شده است :

علاوه بر این ، اگر شما هر بعدی را از قلم بیندازید ، می توانید تمام ستونها یا ردیف های ماتریس را بدست آورید :

توجه داشته باشید که بعنوان نتیجه گیری از یک زیر مجموعه ، ممکن است کار شما به یک بردار ختم شود ، همانطوریکه در دو مثال بالا دیدید . با این همه اگر شما باز هم می خواهید این نتیجه بصورت یک ماتریس باشد ، شما می توانید از فرمول پایین بجای آن استفاده نمایید :

توابع `cbind()` و `rbind()` شاید برای پیوند دو یا تعداد بیشتری ماتریس یا بردار به یکدیگر ، به ترتیب بوسیله ستونها یا ردیف ها بکار روند . مثالهای پایین باید این موضوع را نشان دهد :

شما همگچنین می توانید ستونها و ردیفهای ماتریس ها را ، به ترتیب با استفاده از توابع `colnames()` و `rownames()` ، نامگذاری کنید . این امکانات موقعیتهای این داده را ذخیره می کند .

آرایه ها ، نوع گسترده ماتریس ها برای بیشتر از دو بعد می باشند . این بدان معنی است که آنها دارای بیش از دو شاخص هستند . جدا از اینکه آنها به ماتریس ها شباهت دارند و می توانند به همان روش بکار روند . شبیه به تابع `matrix()` ، یک تابع `array()` برای تسهیل در ایجاد آرایه ها وجود دارد . مورد زیر مثالی از این کاربرد می باشد :

شما می توانید از همان طرحهای شاخص برای دستیابی به عناصر یک آرایه استفاده کنید . مطمئن شوید که مثال زیر را درک کرده اید :

قواعد عملیات حسابی و بازیافت برای ماتریس ها و آرایه ها نیز بکار می رود اگرچه فهمیدن آنها در بعضی اوقات مشکل می شود . در پایین تعدادی مثال آمده است :

R شامل کاربرها و توابع بر ای ماتریس جبری استاندارد نیز می شود که دارای قواعد مختلفی هستند . شاید شما بتوانید اطلاعات بیشتر در این زمینه را با جستجو در بخش 5 این کتاب تحت عنوان « مقدمه ای به R » که همراه R می باشد بدست آورید .

1.2.9 فهرستهای R دربر گیرنده یک مجموعه مدون از اشیای دیگر که تحت عنوان ترکیبات یا اجزای تشکیل دهنده از آنها یاد می شود ، می باشد . بر خلاف عناصر بردارها ، اجزای تشکیل دهنده فهرست نیازی به این ندارد که دارای طول ، مد یا نوشتار یکسانی باشد . اجزای تشکیل دهنده یک فهرست معمولاً شماره گذاری می شوند و همچنین ممکن است دارای یک اسم متصل به خود باشند .

به مثالی در مورد چگونگی ایجاد یک فهرست دقت کنید :

شیء 1st . my بوسیله سه جزء تشکیل دهنده شکل مکی گیرد . اولی یک شماره است و نامش stud . id می باشد ، دومی یک رشته کاراکتر که دارای اسم stud . name می باشد و سومین مورد یک بردار اعداد با نام stud . marks می باشد .

برای نشان دادن محتویات یک فهرست همانطوریکه در مورد اشیاء دیگر انجام می دهید می توانید به سادگی نام محتویات مربوطه را تایپ کنید :

شما می توانید عناصر فردی فهرستها را با استفاده از طرح شاخص زیر ، استخراج کنید :

شاید شما توجه کرده باشید که ما از دو گروه استفاده کرده ایم . اگر ما `my[1]` را بجای آن استفاده کرده باشیم ، نتیجه متفاوتی را بدست می آوریم :

علامت اخیر ، یک زیر فهرست تشکیل شده از اولین جزء تشکیل دهنده `my.1st` را استخراج می کند . در حالت عکس ،

`my[[1]]` ، مقدار اولین جزء تشکیل دهنده را (در این مورد یک عدد) ، که دیگر یک فهرست نیست ، استخراج می کند . همانطور که شما می توانید از طریق زیر آن را اثبات نمایید :

در مورد فهرستهایی با اجزاء تشکیل دهنده نامگذاری شده (همانند مثال قبل) ، ما می توانیم از یک روش جایگزین استخراج مقدار جزء تشکیل دهنده یک فهرست استفاده نماییم :

نامهای اجزاء تشکیل دهنده یک فهرست ، در واقع ، نماد فهرست می باشند و می توانند همانطوری که ما نسبت به نامهای عناصر بردارها ترتیب اثر دادیم ، مورد دستکاری واقع شوند :

فهرستها را می توان با افزودن اجزاء تشکیل دهنده بیشتر به آنها گسترش داد :

شما تعداد اجزاء تشکیل دهنده یک فهرست را با استفاده از تابع (`length (my . 1st)`) می توانید کنترل و بررسی نمایید :

شما می توانید اجزاء تشکیل دهنده یک فهرست را همانند زیر حذف نمایید :

شما می توانید فهرستها را با استفاده از تابع (`C`) به هم پیوند دهید .

در نهایت اینکه شما می توانید تمام داده ها را در یک فهرست با استفاده از تابع (`unlist`) متمرکز کنید . این کار ، برداری را با عناصر زیاد همانطوری که اشیاء داده در یک فهرست وجود دارند ، ایجاد می کند . این کار انواع اطلاعات و داده های مختلف را به یک نوع داده های مشترک تبدیل می کند که بدان معنی است که بیشتر اوقات سر و کار شما با هر چیزی که رشته های کاراکتر محسوب می شوند ، خواهد بود . علاوه بر اینها ، هر عنصر این بردار دارای یک نام برگرفته از نام اجزاء تشکیل دهنده فهرست می باشد که آن را ایجاد می کند .

1.2.10 . چارچوب داده ها

چارچوب داده ها ، ساختار داده ای هستند که در بیشترین حالت نشان دهنده ی جداول داده ای ذخیره شده در `R` می باشند . آنها از لحاظ ساختار شبیه به ماتریس ها هستند همانطوری که آنها دو بعدی هم هستند . به هر حال بر خلاف ماتریس ها ، چارچوب داده ها ممکن است شامل داده های نوع مختلفی در هر ستون باشند . در این معنی ، آنها بیشتر شبیه به فهرستها هستند و در واقع چارچوب داده ها در `R` به مثابه گروه خاصی از فهرستها می باشند . ما می توانیم هر ردیف یک چارچوب داده ها را همانند یک دیدگاه در نظر بگیریم (یا مورد) که بوسیله مجموعه ای از متغیرها تشریح میگردد (ستونهای دارای اسم چارچوب داده) . شما می توانید یک چارچوب داده همانند مورد زیر ایجاد کنید :

توجه داشته باشید که ستون " season " تبدیل به یک ضرب شده است زیرا همه عناصرش رشته های کاراکتری می باشند . از وجه تشابه ، ستون " site " نیز یک ضرب است . این رفتار پیش فرض تابع () frame data می باشد .

شما می توانید از طرح های شاخص مشروح در قسمت 1.2.7 با چارچوب داده ها استفاده نمایید . علاوه بر این شما می توانید از نامهای ستون برای دستیابی به ستونهای کامل یک چارچوب داده استفاده کنید :

شما می توانید چند پرس و جوی ساده را درباره داده های موجود در چارچوب داده ها انجام دهید ، و از امکانات زیر مجموعه ای R همانطوریکه در این مثالها نشان داده است، بهره کامل ببرید :

شما می توانید نوشتن این پرس و جوها را با استفاده از تابع () attach ساده کنید که بدین ترتیب امکان دسترسی شما را به ستونهای یک چارچوب داده بطور مستقیم بدون اینکه مجبور به استفاده از نام چارچوب داده مربوطه شوید ، فراهم کند :

هر زمان که شما به سادگی پرس و جو می کنید در چارچوب داده ، ممکن است در استفاده از تابع () subet ، آن را بطور ساده تری پیدا کنید :

توجه داشته باشید که به هر حال ، بر خلاف مثالهای دیگری که در بالا بررسی شد ، شاید شما از این استراتژی زیر مجموعه برای تغییر دادن مقادیر در داده ها استفاده نکنید . بنابر این بعنوان مثال اگر شما بخواهید 1 را با مقادیر PH تمام ردیف های تابستان (summer) جمع بزنید ، شما فقط به این طریق می توانید آنرا انجام دهید :

شما می توانید ستونهای جدید را ، به همان روشی که در مورد فهرستها انجام دادید ، به یک چارچوب داده بیفزایید :

تنها محدودیتی که در برابر این اضافه کردن وجود دارد این است که ستونهای جدید باید دارای همان شماره ردیفها باشند که در چارچوب داده موجود است ؛ در غیر اینصورت R اعتراض خواهد کرد . شما می توانید شماره ردیف ها یا ستونهای یک چارچوب داده را با این دو تابع ، کنترل و بررسی کنید :

معمولا شما مجموعه داده هایتان را چه بصورت چند فایل و چه بصورت یک پایگاه داده ها ، اشتباها به یک چارچوب داده تعبیر خواهید کرد . شما به ندرت داده ها را با استفاده از تابع () data.frame می توانید همانند بالا تایپ

می کنید ، بویژه در زمینه استخراج یک داده نمونه . در فصلهای آینده با بررسی و مطالعه توضیحات درباره نحوه استخراج داده ها از مطالعات موردی ، خواهید دید که چقدر این نوع داده ها در چارچوب داده ها اهمیت دارند . در هر مورد شما ممکن است بخواهید که راهنمای "ورودی و خروجی داده در R" را که همراه R می باشد جستجو کنید تا تمام امکانات مختلفی را که R دارد بررسی نمایید .

R دارای یک رابط شبیه به یک برنامه صفحه گسترده ساده می باشد که می تواند برای وارد کردن چارچوب داده های کوچک بکار رود . شما می توانید یک چارچوب داده موجود را با تایپ کردن ویرایش کنید .

همانطوریکه نماد اسمها ، یک بردار می باشد ، اگر شما بخواهید نام یک ستون ویژه را تغییر دهید ، می توانید بصورت زیر تایپ کنید :

در نهایت اینکه ، R با برخی از مجموعه داده های توکار همراه می شود که شما می توانید از آنها برای کشف و بررسی برخی از امکانات آن استفاده نمایید . اغلب بسته های نرم افزاری اضافه شونده معمولا همراه مجموعه داده ها می باشند . به منظور کسب اطلاعات در مجموعه داده های موجود تایپ کنید :

برای استفاده از هر مجموعه داده در دسترس ، می توانید به شرح زیر عمل کنید :

این دستور العمل ، یک چارچوب داده را به نام USArrests ایجاد می کند که شامل اطلاعات این مسئله می باشد که به همراه R می آید .

1.2.11 . ایجادتوابع جدید

R این امکان را به کاربر می دهد که توابع جدید ایجاد کند . این یک عملکرد موثر است ، بویژه وقتی که شما بخواهید عملکردهایی را که شما مجبورید بارها و بارها تکرار کنید ، بصورت خودکار درآورید . بجای نوشتن و تایپ دستورالعملهایی که این عملکرد را هر زمان که شما بخواهید اجرا نمایید ، برایتان مهیا سازد ، شما آنها را در یک تابع جدید بطور خلاصه بگنجانید و سپس هر زمان که نیاز احساس کردید به سادگی آنرا بکار ببرید .

توابع R اشیائی همانند این دستورالعملها می باشند که شما در بخشهای پیشین دیده اید . همانند یک شیء ، یک تابع می تواند یک مقدار را ذخیره نماید . این " مقدار " ذخیره شده در یک تابع ، مجموعه دستورالعملهایی است که R در زمانی که شما این تابع را فراخوان می کنید ، آنها را اجرا کند . بنابر این ، برای ایجاد یک تابع

جدید ، شخص ، از کاربر جایگزین برای ذخیره کردن محتویات این تابع با نام یک شیء استفاده می نماید . (نام تابع) .

با یک مثال ساده شروع می کنیم . فرض کنید شما در اغلب موارد می خواهید خطای استاندارد یک رابط میانگین را برای مجموعه ای از مقادیر محاسبه کنید . خطای استاندارد یک میانگین نمونه به طور روشن معین می شود به وسیله جایی که s^2 واریانس نمونه و n اندازه نمونه محسوب می شود .

با توجه به یک بردار مقادیر ، ما یک تابع برای محاسبه خطای استاندارد مربوطه نیاز داریم . نام این تابع را se می گذاریم . قبل از اقدام به ایجاد این تابع ما باید بررسی کنیم که آیا از قبل تابعی به این نام در R وجود دارد یا نه . اگر همانی بود که ما می خواهیم ، پس بهتر آن است که با نام دیگری از آن استفاده نماییم . نه اینکه تابع R دیگری را از دید کاربر مخفی کنیم . برای بررسی موجود بودن آن تابع ، کافی است که نام آنرا در فراخوان تایپ کنید :

خطای چاپ شده بوسیله R نشاندهنده ی این است که ما مجاز به استفاده از آن نام هستیم . اگر یک تابع (یا هر شیء دیگری) با نام " se " وجود داشته باشد ، R محتوای آنرا به جای خطا چاپ خواهد کرد .

توضیحات زیر راه ساده ای را برای ایجاد تابع مد نظر ما نشان می دهد :

بنابر این ، برای ایجاد یک شیء تابع ، شما اختصاص بدهید به نام آن چیزی را که شکل کلی داشته باشد .

بعد از ایجاد این تابع ، شما می توانید آنرا همانند شرح زیر استفاده نمایید :

اگر ما نیاز به به اجرای چند دستورالعمل با ابزار یک تابع داشته باشیم ، همانند موردی که برای تابع (se) انجام دادیم ، ما نیاز به شکلی تازه گفتار R داریم زمانیکه شکل تابع شروع می شود و زمانیکه شکل «پایان می یابد» . R از آکلاک بعنوان عناصر ساختار دستور که گروهی از دستورالعملها را شروع می کند و دوباره به آنها خاتمه می دهد ، استفاده می نماید . مقدار برگشت داده شده بوسیله هر تابع می توان با استفاده از تابع ($return$) تصمیم گیری کرد ، یا به جای آن ، R نتیجه عبارت آخر را برمی گرداند که در داخل تابع ارزیابی شده است . تابع زیر این مورد را نشان می دهد و همچنین نشاندهنده استفاده از پارامترها یا مقادیر پیش فرض می باشد .

این تابع دارای یک پارامتر (more) می باشد که دارای یک مقدار پیش فرض (F) می باشد . این بدان معنی است که شما می توانید این تابع را با قرار دادن این پارامتر یا بدون قرار دادن آن فراخوان نمایید . اگر شما آنرا بدون یک مقدار برای پارامتر دوم فراخوان کنید ، مقدار پیش فرض ، مورد استفاده قرار خواهد گرفت . دو مثال در مورد این دو گزینه در پایین آورده شده است :

تابع (basic.stats) نیز یک دستورالعمل جدید از R را معرفی می نماید :

دستورالعمل (if) . همانطوریکه از نام این دستورالعمل پیداست این امکان را به ما می دهد تا اجرای دستورالعمل های معین را برای مقدار حقیقی یک آزمایش منطقی ، شرطی کنیم . در مورد این تابع ، دو دستورالعمل که اوج و انحراف تابع مقادیر را محاسبه می کنند ، تنها در صورتی اجرا می شوند که متغیر more حقیقی باشد ، در غیر این صورت از اجرای آنها صرف نظر می شود .

دستورالعمل مهم دیگر ، (orf) می باشد . این دستورالعمل ما را قادر می سازد تا مجموعه ای از فرمان ها را چند بار تکرار کنیم . در پایین مثالی برای استفاده از این دستورالعمل آمده است :

برای فراخوان (f (5)) دستورالعمل موجود در این تابع به R می گوید که دستورالعملهای "داخل آن را " چند بار باید اجرا شوند (به وسیله آکلاد محصور شده اند). بدین معنی که ، آنها باید به وسیله متغیر " i " که در هر تکرار ، مقدار متفاوتی را بپذیرد ، اجرا شوند . در این مثال " i " مقادیری را در مجموعه 1:10 ، قبول کند که عبارتند از 1,2,3,....,10 . این بدان معنی است که دستورالعمل داخل for ، ده بار و هر بار با مجموعه i که دارای یک مقدار متفاوتی است ، اجرایی شوند . مجموعه مقادیر مشخص شده در جلوی کلمه in می توانند هر برداری باشند ، و نیازی نیست که مقادیر به صورت توالی یا عددی باشند . تابع (cat) می تواند برای خروجی محتویات چندین شیء با این حالت از نمایش بکار روند . بدین معنی که ، رشته های کاراکتر ، همانند محتوای خودشان تایپ می شوند ('hello' cat () را امتحان کنید) ، در حالیکه همانند محتوای خودشان تایپ می شوند (ابتدا امتحان کنید y <-45 و سپس (y cat () را امتحان نمایید) . رشته " \n " ، R را به چند خط تبدیل می کند .

1.2.12. اشیاء ، طبقه بندی ها و روشها

یکی از اهداف طراحی R آسان سازی دستکاری در داده ها و اطلاعات می باشد به طوریکه مامی توانیم به راحتی تکالیف تجزیه و تحلیل داده ای خودمان را انجام دهیم . در R ، داده ها در اشیاء ذخیره می شوند . همانطوریکه اشاره گردید ، هر چیزی در R ، از اعداد ساده گرفته تا توابع یا ساختارهای داده ای پیچیده تریک

شیء را در برمی گیرد. هر شیء R به یک طبقه بندی تعلق دارد. طبقه بندیها، ویژگیهای انتزاعی اشیاء را که به آنها تعلق دارند، تعریف می نمایند. بدین معنی که آنها علایم یا ویژگیهای این اشیاء و حتی عکس عملهای آنها را (یا روشهای آنها را) مشخص می کنند. مثلاً، طبقه بندی ماتریکس دارای ویژگیهای خاصی مانند بعد این ماتریسها و عکس عملهای معین برای چندنوع از عملیات می باشد. در واقع، ما از R ، محتوای یک ماتریکس را درخواست می کنیم، R آنرا با فرمت خاصی در این صفحه به ما ارائه خواهد کرد. دلیل این اتفاق به این خاطر است که یک روش چاپ خاص در مورد تمام اشیاء ماتریکس این طبقه بندی وجود دارد. خلاصه اینکه، طبقه بندی یک شیء (1) روشهایی را که به وسیله چند تابع کلی هنگامی که در ارتباط با این اشیاء بکار می روند و همچنین (2) نمایش اشیاء آن طبقه بندی را مشخص می کند.

R دارای طبقه بندیهای از قبل مشخص شده فراوانی برای اشیاء، به همراه روشهای مرتبط با آنها می باشد. بعلاوه اینکه ما می توانیم این فهرست را به وسیله ایجاد طبقه بندیهای جدید اشیاء یا روشهای جدید، توسعه دهیم. این دوروش جدید می توانند هر دو در این طبقه بندیهای جدید به طور طبیعی بعد از طبقه بندیهای موجود، معمولاً به وسیله اضافه کردن چند بخش جدید اطلاعات، به محتویات آنها، ایجاد شوند. محتویات یک طبقه بندی در برگیرنده مجموعه ای از اسلات ها می باشد. هر اسلات دارای یک اسم و یک طبقه بندی وابسته است. اطلاعاتی را که ذخیره می کند، مشخص می کند. کاربر " l " می تواند برای دستیابی به اطلاعات ذخیره شده در اسلات یک شیء بکار رود. این بدان معنی است که YlX ، مقدار اسلات Y یک شیء X می باشد. این به طور طبیعی فرض می انگارد که طبقه بندی اشیایی که به X تعلق دارند دارای یک اسلات اطلاعات به نام Y می باشد. نظریه مهم دیگر راجع به طبقه بندیها، نظریه وراثت موجود بین طبقه بندیهاست. این دیدگاه، روابطی را مابین این طبقه بندیها ایجاد می کند که به ما اجازه می دهد تا نشان دهیم که یک طبقه بندی مشخص جدید، یک طبقه بندی موجود را به وسیله اضافه کردن مقداری اطلاعات جدید، بسط می دهد. این بسط و گسترش از طرفی حاکی از آن است که این طبقه بندی جدید، تمام روشهای طبقه بندی قبل را به ارث می برد. ف که ایجاد طبقه بندی های جدید را آسان می کند، چرا که ما دوباره از اول شروع نمی کنیم. در این متن لازم است که ما فقط نگران اجرای روشهایی برای انجام این عملیات باشیم جایی که طبقه بندی جدید اشیاء با طبقه بندی موجودی که به وسیله طبقه بندی جدید توسعه می یابد، فرق دارد.

در نهایت اینکه، نظریه خیلی مهم دیگر، نظریه پلی مورفیسم یا چندشکلی می باشد. این نظریه ثابت می کند که برخی توابع می توانند با طبقه بندی های مختلف اشیاء بکار روند و نتایجی را که برای طبقه بندی مربوطه، مناسب می باشند ف ایجاد نمایند.

در R، این نظریه قویا با نظریه تابع مولد درارتباط است. توابع مولد، یک عملکرد سطح بالی مشخص بسیار کلی را اجرا می کنند به عنوان مثال، همانطوریکه ما خواهیم دید،

تابع () plot می تواند برای به دست آوردن محتوای گرافیکی یک شیء بکار رود.

این همان هدف کلی آن است. به هر حال، این محتوای گرافیکی، واقعاً ممکن است بر مبنای نوع شیء متفاوت باشد. بعنوان مثال، رسم مجموعه ای از اعداد با رسم یک مدل همبستگی خطی با هم فرق دارند. چند شکلی، کلیدی برای حل این مسئله بدون اینکه ذهن کاربر را مشوش کند، می باشد. کاربر فقط نیاز است که بداند که تابعی وجود دارد که یک محتوای گرافیکی را برای اشیاء فراهم می کند. R و مکانیسم های داخلی آن، بر عمل ارسال این تکالیف کلی برای توابع خاص این طبقه بندی مدیریت می کنند که این توابع، محتوای گرافیکی را برای هر طبقه بندی اشیاء فراهم می کند.

کل فرایند این روش ارسال، بدون اینکه کاربر در این زمینه، از جزئیات نادرست آن خبر داشته باشد، اتفاق می افتد. در واقع آن چیزی که اتفاق می افتد، این است که چون R می داند که () plot یک تابع مولد است، پس برای یک روش رسم کردن (plot) که مخصوص برای طبقه بندی اشیایی که تحت فراخوان تابع () Plot قرار گرفته اند، محسوب می شود، جستجو را آغاز خواهد کرد، اگر چنین روشی وجود داشته باشد، R از آن استفاده خواهد کرد. در غیر این صورت به بعضی از روشهای رسم پیش فرض مراجعه خواهد کرد. زمانی که کاربر تصمیم می گیرد تا طبقه بندی جدیدی از اشیاء را ایجاد کند، او مجبور است تصمیم بگیرد اگر که بخواهد توان رسم کردن اشیاء این طبقه بندی جدید را داشته باشد، آنگاه او مجبور است روش رسم ویژه ای را برای این طبقه بندی جدید اشیاء فراهم نماید تا به R بگوید که چگونه اشیاء جدید را رسم نماید.

اینها ریزه کاری های بسیار هم طبقه بندی ها و روش ها در R می باشند. ایجاد طبقه بندی های جدید و روشهای وابسته از حوزه بحث این کتاب خارج است.

اطلاعات بیشتر در این باره را می توانید در بسیاری از کتابهای موجود در باب برنامه نویسی با R، مثل کتاب ارزشمند نرم افزار تجزیه و تحلیل داده که نوشته چمبرز (2008) می باشد.

زمانی که شما از R برای تکالیف پیچیده تر استفاده می کنید، این کار باعث محدود شدن خط فرمان می شود که نوع تعامل و ارتباط را تایپ می کند. در این وضعیت ها، معقول تر آن است که کدتان را در یک فایل متنی بنویسید و سپس از R بخواهید آنرا اجرا نماید.

برای ایجاد چنین فایلی شما می توانید از ویراشگر متنی مورد علاقه تان (مثل نوت پد، ایماکس و غیره) استفاده نمایید، یا در موردی که شما از نگارش ویندوز R استفاده می کنید شما می توانید از ویرایشگر اسکریپت (دست نویس) موجود در منوی این فایل بهره بگیرید. پس از ایجاد و ذخیره این فایل، شما می توانید فرمان زیر را در فراخوان R صادر نمایید تا تمام فرمانهای موجود در این فایل را اجرا کند:

این فرمان فرض می کند که شما دارا یک فایل متنی به نام mycod.R در فهرست راهنمای کار جاری R می باشد. در نگارشهای ویندوز، آسان ترین راه برای تغییر این فهرست راهنما، از طریق گزینه `change directory` مربوط به منوی `file` می باشد. در نگارشهای یونیکس، ممکن است شما از توابع `(getwd)` و `(setwd)` به ترتیب برای کنترل و تغییر فهرست راهنمای کار جاری، استفاده نمایید.

زمانی که شما از فراخوان R در یک مدل تعاملی استفاده می کنید شما می خواهید بعضی از اشیایی را که برای استفاده آتی ایجاد کرده اید را ذخیره نمایید (مثلاً چندتابع تایپ کرده باشید). مثال زیر اشیایی به نام `f` و `my.dataset` را در فایلی به نام `mysession.RData` ذخیره می کند.

بعدها، مثلاً در یک جلسه جدید R، شما می توانید این اشیاء را با صادر کردن فرمان زیر لود کنید:

همچنین می توانید تمام اشیاء را هم اکنون در فضای کار R ذخیره کنید. این کار با صدور فرمان زیر صورت می گیرد.

این فرمان، این فضای کار را در فایلی به نام `RData` در فهرست راهنمای کار جاری ذخیره می کند. این فایل به طور خودکار زمانی که شما R را دوباره از این فهرست راهنما اجرا می کنید، لود می شود. به این نوع نتایج می توان با جواب دادن بله در زمان خارج شدن از R نایل شد. (به قسمت 1-2- بنگرید).

مطالعات بیشتر در مورد

راهنمای آنلاین مقدمه ای بر R که با هر قسمت از R همراه است یک منبع ارزشمند اطلاعاتی در مورد زبان برنامه نویسی R می باشد. زیر بخش Contributed در قسمت Documentation در وب سایت R، چند کتاب آزاد را در باب جنبه های مختلف R، ارائه می نماید.

3-1- مقدمه کوتاهی بر MySQL به دست می دهد. MySQL خیلی ضرورت ندارد تا تمام مطالعات موردی این کتاب را انجام دهید. با وجود این، برای پروژه های استخراج اطلاعات بزرگتر، استفاده از یک سیستم مدیریت پایگاه داده همانند MySQL می تواند حیاتی باشد.

<http://www.mysql.com>

MySQL می تواند بدون پرداخت هیچ هزینه ای از وب سایت دانلود شود. همانند R، MySQL برای سیستم های عامل مختلف مثل لینوکس و ویندوز، موجود است. اگر شما خواسته باشید MySQL را بر روی کامپیوترتان نصب کنید، شما باید آن را از وب سایت MySQL را بر روی کامپیوترتان نصب کنید، شما باید آن را از وب سایت MySQL دانلود کنید و از دستورالعمل های نصب آن پیروی کنید. یا اینکه به جای آن، شما همچنین می توانید به هر سرویس دهنده MySQL دسترسی پیدا کنید که در کامپیوتر دیگری که شما دسترسی به شبکه دارید قابل نصب است.

شما می توانید از یک برنامه سرویس گیرنده برای به دست آوردن MySQL در کامپیوتر محلی تان یا در داخل اینترنت، استفاده نمایید. تعداد بسیار زیادی برنامه سرویس گیرنده MySQL در وب سایت MySQL وجود دارد. MySQL به همراه یک برنامه سرویس گیرنده از نوع کنسول ظاهر می شود که با یک مدل فرمان به فرمان، همانند کنسول R عمل می کند. یا اینکه شما دارای برنامه های سرویس گیرنده گرافیکی که می توانید برای استفاده در MySQL آنها را نصب نمایید. بویژه، جستجوگر کندوکاوی MySQL به آسانی در دسترسی است و تقریباً نمونه خیلی خوبی از این برنامه هاست که شما ممکن در نظر داشته باشید در کامپیوتر خود آن را نصب نمایید.

برای دستیابی به سرویس دهنده MySQL نصب شده در کامپیوترتان، با استفاده از سرویس گیرنده از نوع کنسول، شما می توانید فرمان زیر را در فراخوان سیستم عامل تان صادر کنید:

یا در مورد یک سرویس دهنده از راه دور، چیزی مثل فرمول زیر را صادر نمایید:

ما فرض می کنیم که این سرویس دهنده دارای یک کاربر به نام Myuser و اینکه، این سرویس دهنده دارای پس ورد محافظ می باشد. اگر تمام این موارد برای شما عجیب به نظر برسد، شما باید با مدیر سیستم تان در رابطه با MySQL صحبت کنید، یا کمی بیشتر در باره این نرم افزار بیاموزید با استفاده از دفترچه راهنمای کاربر که با هر نصب همراه است، یا اینکه یک کتاب در این مورد مطالعه کنید (بعنوان مثال 2000DuBois)

پس از وارد شدن به MySQL، شما می توانید یا از پایگاه داده موجود استفاده کنید یا پایگاه داده جدیدی را ایجاد کنید. مورد دوم همانطوری که در سرویس گیرنده از نوع کنسول MySQL می توانند به شرح زیر انجام شود:

برای استفاده از پایگاه داده به تازگی ایجاد شده یا هر پایگاه داده موجود دیگر، شما فرمان زیر را صادر کنید:

یک پایگاه داده بوسیله مجموعه ای از جداول در بردارنده داده های راجع به برخی موجودات تشکیل می شود. شما می توانید یک جدول مثل زیر ایجاد کنید.

توجه داشته باشید به فراخوان ادامه دار MySQL ("`>`")

برای پر کردن یک جدول با داده ها، شما می توانید و ثبت را بطور دستی وارد کنید یا از یکی از جمله های وارد کردن MySQL برای خواندن داده های موجود در جدول بعنوان مثال در یک فایل متنی، استفاده نمایید

یک مثبت یا گزارش می تواند مثل زیر در یک جدول وارد شود: `SELECT` شما می توانید مثبت ها را در یک جدول مشخصی با استفاده از جمله فهرست نمایید که ما چند مثال در پایین آورده ایم:

بعد از اینکه شما کارتان با MySQL به پایان رسید شما می توانید سرویس گیرنده از نوع کنسول را با صادر کردن جمله ی `quit` ترک نمایید.

پیش بینی رشد انفجاری جلبک دریایی

این بررسی موردی شما را با برخی وظایف اساسی اطلاعات استخراج معدن آشنا می کند: اطلاعات پیش پردازش شده، تحلیل

اطلاعات اکتشافی، و تفسیر مدل پیشگویانه برای این بررسی موردی ابتدایی مشکل کوچکی را به واسطه اطلاعات معدنی انتخاب

کرده ایم. برای مثال، به مشکل فروانی رشد بسیاری از خزّه های مضر در آب های نمونه اشاره می کنیم. اگر با زبان R آشنا نباشید

یا در بخش 1.2 در فصل 1 نخوانده باشید لازم است که این بخش را دوباره مروری کنید.

2.1 مشکل توصیف و اهداف

تمرکز بسیار بر خزّه های مضر در رودخانه ها باعث ایجاد مشکلات اکولوژیکی با تاثیر قوی هم بر نوع زندگی رودخانه ها شده

و هم بر کیفیت آب ها. قادر بودن به کنترل و اعمال پیش بینی سریع در باره رشد جلبک ها برای توسعه کیفیت رودخانه ها ضروری

است.

با داشتن هدف اشاره به این مشکل پیش بینی، بسیاری از نمونه های آب در رودخانه های مختلف اروپا و در زمان های متفاوت به

مدت یک سال جمع آوری شده است. برای هر نمونه آب مواد شیمیایی متفاوتی به ازای فراوانی رشد هفت خزہ مضر به کار برده شده

است. برخی از مشخصات اصلی دیگر پردازش مجموعه آب هم مثل فصل سال، سایز رودخانه و سرعت رودخانه ذخیره شده است.

یکی از محرک های این عمل در این است که کنترل شیمیایی ارزان تر و به کارگرفتنش اسان تر است؛ در حالیکه تحلیل بیولوژیکی

نمونه ها به منظور شناسایی خزہ هایی که در آب موجودند نیازمند آزمایش های میکروسکوپی، نیروی کار آزموده است و هر دوی

آن ها گران و کند است. همین طور دست یافتن به مدل هایی که قادر به پیش بینی دقیق فراوانی های خزہ ها بر اساس مواد شیمیایی

هستند تکوین سیستم های خودکار و ارزان را برای کنترل رشد خزہ های مضر تسهیل میکند.

هدف دیگذا این بررسی ایجاد امکان درک بهتر عوامل موثر بر فراوانی های خزہ ها است. برای مثال، می خواهیم چگونگی ارتباط این

فراوانی ها را با ویژگی های شیمیایی نمونه ها آب همانند سایر خصوصیات این نمونه ها بدانیم (مثل فصول سال، نوع رودخانه و...).

2.2 توصیف اطلاعات

این اطلاعات برای این مشکل در متن تحقیقات ERUDIT جمع آوری شده و در مسابقات COIL سال 1999 مورد استفاده قرار گرفته

شده قابل دسترس است. از منابع بسیاری مانند مخزن یاد گیری ماشین UCI قابل دسترس است.

دو مجموعه داده برای این مشکل وجود دارد. اولی شامل داده هایی برای 200 نمونه آب ها است. با دقت بیشتر، هر مشاهده در دادهای

قابل دسترس تحت تاثیر انبوهی از نمونه های آبی بسیار است که از رودخانه یکسان طی دوره ای سه ماهه و در فصلی یکسان جمع

آوری شده است.

هر مشاهده ای شامل اطلاعاتی درباره 11 متغیر است. سه مورد از این متغیرها اسمی هستند و فصول سال را توصیف در زمانی

که آب های نمونه به صور متراکم جمع شده است را مانند سایز و سرعت رودخانه در سوال توصیف می کنند. هشت متغیر باقی مانده

به مقدار های متفاوت پارامترهای شیمیایی در آب نمونه برداری شده به صورت متراکم اندازه گیری میشود. مانند:

- Maximum pH value
- Minimum value of O_2 (oxygen)
- Mean value of Cl (chloride)
- Mean value of NO_3^- (nitrates)
- Mean value of NH_4^+ (ammonium)
- Mean of PO_4^{3-} (orthophosphate)
- Mean of total PO_4 (phosphate)
- Mean of chlorophyll

در مقایسه با هر یک از این پرامترها به به مقدار متفاوتی خزه مضر در آب نمونه برداری شده مربوطه یافت شده است. هیچ اطلاعاتی

راجع به اسم خزه های یافت شده وجود ندارد. داده دومی شامل اطلاعاتی راجع به 140 مشاهده بیشتر می شود. این هم از ساختاری

یکسان استفاده می کند اما شامل اطلاعات درگیر با فراوانی خزه های خطرناک نمی شود. این مشاهدات اضافی می تواند به عنوان

یک نوع از داده های آزمایشی تلقی گردد. هدف اصلی از این مطالعات پیش بینی فراوانی های هفت خزه برای این 140 نمونه آب

است. این بدین معناست که با داده استخراجی مربوطه مواجه هستیم. این یکی از وظایف میان مجموعه ای از مشکلات متنوع موجود

در داده های استخراجی است. در این نوع از تکالیف هدف اصلی ما رسیدن به مدلی است که به ما اجازه بدهد که مقدار متغیر اصلی

را پیش بینی کنیم. این مدل قوانینی را بر این که کدام متغیر احتمالی دارای تاثیر بیش تری بر متغیر اصلی است دارد. که به این معنی

است که این مدل توصیف ادراکی از عواملی که بر متغیر اصلی تاثیر دارد تهیه می کند.

2.3 ذخیره کردن داده ها در R

به دو نمونه از شکل های ذخیره داده اها در R اشاره میکنیم: (1) با استفاده از پکیج همراه با کتاب که شامل صفحه اطلاعاتی و داده

های قابل استفاده (2) با استفاده از مراجعه به سایت کتاب و دانلود فایل های متنی همراه با داده ها و سپس ذخیره آن ها در R. راه

قبلی عملی تر است. اطلاعات را در جایگزین دومی برای هدف تصویری در مورد چگونگی ذخیره داده ها بر R قرار می دهیم.

اگر خواهان پیروی از این راه آسان هستید به سادگی ابتدا پکیج را ذخیره کنید و به سرعت دارای یک صفحه اطلاعاتی به نام خز

قابل دسترس برای استفاده خواهید بود. این داده ها شامل اولین مجموعه 200 مشاهده که در زیر به آن اشاره شده است می شود.

```
> library(DMwR)
> head(algae)
```

	season	size	speed	mxPH	mnO2	Cl	N03	NH4	oP04	P04	Chla
1	winter	small	medium	8.00	9.8	60.800	6.238	578.000	105.000	170.000	50.0
2	spring	small	medium	8.35	8.0	57.750	1.288	370.000	428.750	558.750	1.3
3	autumn	small	medium	8.10	11.4	40.020	5.330	346.667	125.667	187.057	15.6
4	spring	small	medium	8.07	4.8	77.364	2.302	98.182	61.182	138.700	1.4
5	autumn	small	medium	8.06	9.0	55.350	10.416	233.700	58.222	97.580	10.5
6	winter	small	high	8.25	13.1	65.750	9.248	430.000	18.250	56.667	28.4

	a1	a2	a3	a4	a5	a6	a7
1	0.0	0.0	0.0	0.0	34.2	8.3	0.0
2	1.4	7.6	4.8	1.9	6.7	0.0	2.1
3	3.3	53.6	1.9	0.0	0.0	0.0	9.7
4	3.1	41.0	18.9	0.0	1.4	0.0	1.4
5	9.2	2.9	7.5	0.0	7.5	4.1	1.0
6	15.1	14.6	1.4	0.0	22.5	12.6	2.9

یک صفحه اطلاعاتی می تواند به عنوان نوعی ا ز ماتریس یا جدول همراه با ستون های اسمی که داده ایست با ساختار ایده آل

برای نگهداری جدول های اطلاعاتی در R دیده شود. علامت ()شش خط اولی هر صفحه اطلاعاتی را نشان می دهد .

به صورت جایگزین ممکن است از فایل های متنی در بخش داده استفاده کنید. لینک داده های تعلیمی شامل 200 نمونه آب ها

در فایلی به نام "متن تحلیلی" می شود اما لینک داده های آزمایشی به فایل "متن ارزیابی شده" اشاره دارد که شامل 140 نمونه

آزمایشی می شود. یک لینک اضافی دیگر هم وجود دارد که به فایل "متن تحولی" که شامل 140 نمونه آزمایش شده است اشاره

دارد. این فایل آخری برای کنترل عملکرد مدل های اسمی مورد استفاده قرار می گیرد و به عنوان اطلاعات ناشناخته برای حال

در نظر گرفته می شود. این فایل ها در خطی متفاوت دارای مقادیری در هر مشاهده هستند. هر خط از فایل های تعلیمی و تستی

حاوی مقادیری از متغیرهای جدا شده به واسطه فضا می شود (بر اساس توصیف داده شده در بخش 2.2). مقادیر ناشناخته به

شکل "XXXXXXX" نشان داده شده است.

اولین کار آن است که سه فایل را از سایت دانلود کرده و بر روی هارد خود ذخیره کنید.

پس از دانلود فایل داده ها بر روی راهنما می توانیم با ذخیره داده ها از متن تحلیلی بر روی R شروع کنیم (داده های تعلیمی برای

مثال داده هایی که برای رسیدن به مدل های اسمی به کار گرفته می شوند). برای خواندن داده ها از روی فایل کافی است که دستور

زیر را اجرا کنیم:

```
> algae <- read.table('Analysis.txt',
+                   header=F,
+                   dec='.',
+                   col.names=c('season','size','speed','mxPH','mnO2','Cl',
+                   'NO3','NH4','oPO4','PO4','Chla','a1','a2','a3','a4',
+                   'a5','a6','a7'),
+                   na.strings=c('XXXXXXX'))
```

پارامتر header=F بیان می کند که فایل در حال خوانده شدن شامل اولین خط با نام های متغیر ها نمی شود. 'dec='.' بیان می کند

که ارقام از '.' برای جدا کردن فضا های اعشاری استفاده می کنند. این دو پارامتر فبلی در حین آن که از مقادیر قراردادی آن ها

استفاده می کنیم می توانند حذف شوند. col.names به ما اجازه می دهد که برداری با نام های برای دادن به متغیر هایی که ارزش

آن ها خوانده شده تهیه کنیم. در نهایت na.strings در جهت بیان بردار رشته هایی که به عنوان مقادیر ناشناخته در حال تفسیرند

خدمت می کند. این مقادیر به واسطه مقدار NA در R همانطور که در بخش 1.2.3. گفته شده ارائه شده است.

R بسیاری شرایط دیگر هم دارد که می تواند برای خواندن داده ها در فایل متنی مورد استفاده قرار بگیرد. شاید شما بخواهید

"?read.table" را برای به دست آوردن سایر اطلاعات بعدی و سایر شرایط مربوطه تایپ کنید. همچنین R دارای راهنمایی

است که شاید شما بخواهید با این نام "داده های صادراتی/وارداتی" به آن سری بزنید. که احتمالات متفاوت R را که شامل داده

های در خواندن از سایر دستورالعمل ها می شود.

نتیجه راهنمای بالا صفحه اطلاعاتی است. هر یک خطوط این صفحه اطلاعات شامل مشاهده ای از مجموعه داده های ما است. برای

مثال می توانیم با استفاده از عملکرد `alga[1:5]` مشاهده ابتدایی را ببینیم. در بخش 1.2.7 راه های جایگزین استخراج عناصر ویژه R را مانند صفحه اطلاعات توصیف کرده ایم.

2.4 تصویر سازی و خلاصه سازی داده ها

با توجه به فقدان اطلاعات ثانوی این مشکل جست و جوی برخی خواص آماری داده ها به منظور رسیدن به راه حلی بهتر برای

این مشکل عاقلانه به نظر می رسد. حتی اگر این روش موثر نباشد خوب است همیشه تحلیل خود را با برخی داده ها ی تحلیلی

اکتشافی مشابه با آن چه در پایین داریم شروع کنیم.

اولین ایده خواص آماری داده ها از طریق خلاصه ای از آمار توصیفی آن قابل دسترس است:


```
> summary(algae)
```

season	size	speed	mxPH	mnO2
autumn:40	large :45	high :84	Min. :5.600	Min. : 1.500
spring:53	medium:84	low :33	1st Qu.:7.700	1st Qu.: 7.725
summer:45	small :71	medium:83	Median :8.060	Median : 9.800
winter:62			Mean :8.012	Mean : 9.118
			3rd Qu.:8.400	3rd Qu.:10.800
			Max. :9.700	Max. :13.400
			NA's :1.000	NA's : 2.000

C1	N03	NH4	oP04
Min. : 0.222	Min. : 0.050	Min. : 5.00	Min. : 1.00
1st Qu.: 10.981	1st Qu.: 1.296	1st Qu.: 38.33	1st Qu.: 15.70
Median : 32.730	Median : 2.675	Median : 103.17	Median : 40.15
Mean : 43.636	Mean : 3.282	Mean : 501.30	Mean : 73.59
3rd Qu.: 57.824	3rd Qu.: 4.446	3rd Qu.: 226.95	3rd Qu.: 99.33
Max. :391.500	Max. :45.650	Max. :24064.00	Max. :564.60
NA's : 10.000	NA's : 2.000	NA's : 2.00	NA's : 2.00

P04	Chla	a1	a2
Min. : 1.00	Min. : 0.200	Min. : 0.00	Min. : 0.000
1st Qu.: 41.38	1st Qu.: 2.000	1st Qu.: 1.50	1st Qu.: 0.000
Median :103.29	Median : 5.475	Median : 6.95	Median : 3.000
Mean :137.88	Mean : 13.971	Mean :16.92	Mean : 7.458
3rd Qu.:213.75	3rd Qu.: 18.308	3rd Qu.:24.80	3rd Qu.:11.375
Max. :771.60	Max. :110.456	Max. :89.80	Max. :72.600
NA's : 2.00	NA's : 12.000		

a3	a4	a5	a6

Iran

Min. : 0.000	Min. : 0.000	Min. : 0.000	Min. : 0.000
1st Qu.: 0.000	1st Qu.: 0.000	1st Qu.: 0.000	1st Qu.: 0.000
Median : 1.550	Median : 0.000	Median : 1.900	Median : 0.000
Mean : 4.309	Mean : 1.992	Mean : 5.064	Mean : 5.964
3rd Qu.: 4.925	3rd Qu.: 2.400	3rd Qu.: 7.500	3rd Qu.: 6.925
Max. : 42.800	Max. : 44.600	Max. : 44.400	Max. : 77.600

a7

Min. : 0.000
1st Qu.: 0.000
Median : 1.000
Mean : 2.495
3rd Qu.: 2.400
Max. : 31.600

این ساختار ساده به سرعت اولین نظریه را در مورد خواص آماری به ما می دهد. در رابطه با متغیر های اسمی (که به واسطه

عناصر موجود در صفحه اطلاعات R ارائه میگردد) برای هر یک از مقادیر ممکن فراوانی تهیه می کند. برای مثال ما مشاهده

می کنیم که نمونه ها آب ها در زمستان بیش از سایر فصول دیگر جمع آوری شده است. برای متغیر های عددی R به ما مجموعه ای

از آمار ها نظیر معنی، میانگین، چارک ها و مقادیر را به ما می دهد. این آمارها اولین ایده توزیع مقادیر متغیرها را فراهم می کند.

در رابطه با متغیر های دارای برخی ارزش های ناشناس شماره هایشان به صورت زیر آمده است. با مشاهده تفاوت میان میانگین ها

و معانی مانند چارک ها می توانیم به نظریه چولگی توزیع و همچنین پخش آن برسیم. هنوز هم بیشتر وقت ها این اطلاعات به صورت

گرافیکی بهتر محاصره می گردد. بیایید مثالی ببینیم:

```
> hist(algae$mxPH, prob = T)
```

این ساختار هیستوگرام متغیر mxPH را به ما نشان می دهد. نتیجه در شکل 2.1 مشاهده می شود. با پارامتر $prob=T$ به احتمالات

برای هر یک از مقادیر می رسم، اما حذف این پارامتر فراوانی هایی به ما می دهد.

تصویر 2.1 می گوید که مقادیر متغیر mxPH ظاهراً توزیعی بسیار نزدیک به توزیع نرمال را دنبال می کند. یک کنترل دقیق تر

این فرضیه می تواند با استفاده از نقاط نرمال Q-Q به دست آید.

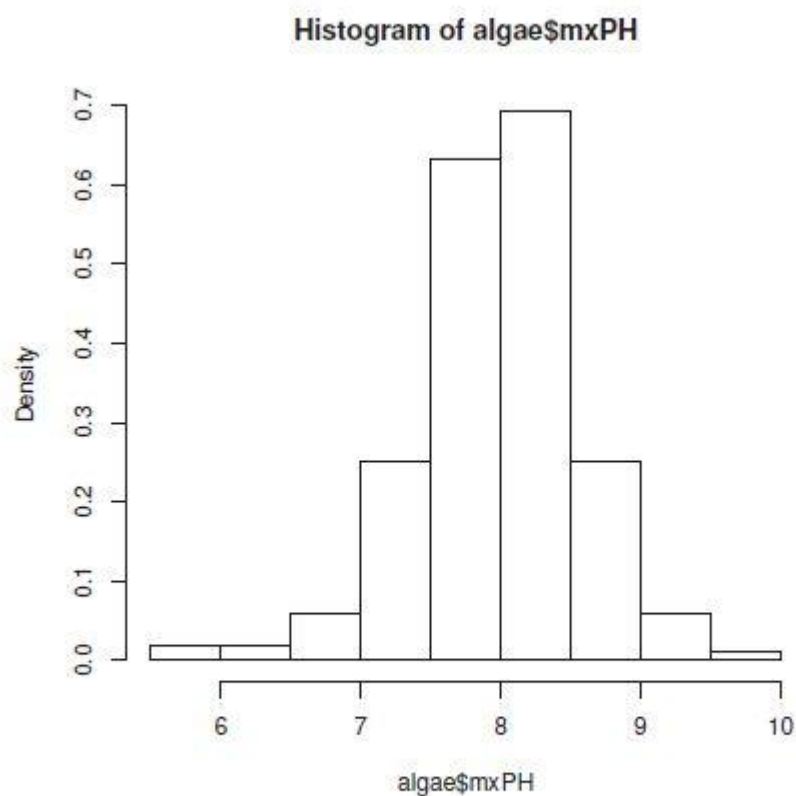


FIGURE 2.1: The histogram of variable $mxPH$.

وضعیت () qq.plot در پکیج car این نقطه را به دست می دهد و نتیجه آن در تصویر 2.2 با سری ماهر هیستوگرام نمایش داده

شده است. نمودار ها به وسیله کد زیر طراحی شده اند:

```

> library(car)
> par(mfrow=c(1,2))
> hist(algae$mxPH, prob=T, xlab='',
+      main='Histogram of maximum pH value',ylim=0:1)
> lines(density(algae$mxPH,na.rm=T))
> rug(jitter(algae$mxPH))
> qq.plot(algae$mxPH,main='Normal QQ plot of maximum pH')
> par(mfrow=c(1,1))

```

پس از ذخیره پکیج کد با نامی از `par()` که می تواند برای ایجاد بسیاری از پارامترهای سیستم گرافیک های R استفاده شود

شروع می شود. در این مورد ما گرافیک ها را به پنجره ای بیرونی در یک خط با دو ستون با هدف زسیدن به دو گرافیک در کنار

هم تقسیم می کنیم. سپس به اولین نمودار که باز هم هیستوگرامی از متغیر `mxPH` است میرسیم، به جز این که این بار مدار `X` را

اشغال کرده ایم، عنوان گراف را تغییر داده و محدودیت های دیگری برای مدار `Y` ایجاد می کنیم.

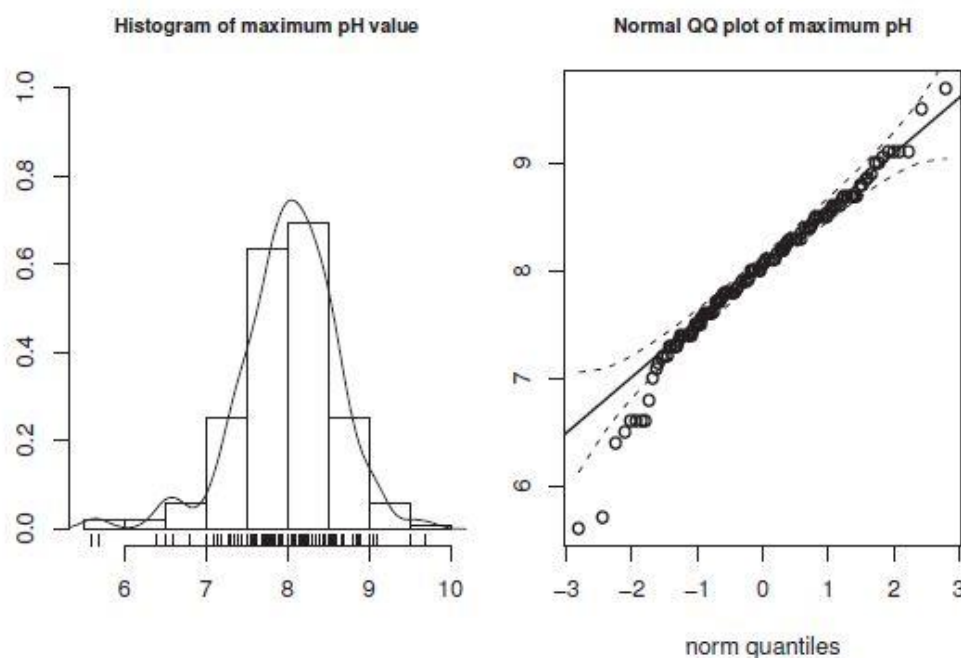


FIGURE 2.2: An “enriched” version of the histogram of variable $MxPH$ (left) together with a normal Q-Q plot (right).

ساختار بعدی سری همواری از هیستوگرام را ایجاد می کند (برآورد تراکم هسته ای توزیع متغیر) اما نقاط زیر مقادیر اصلی

متغیر نزدیک به مدار X را ایجاد میکند. برای مثال مشاهده می کنیم که دو مقدار کم تر از سایر مقادیر وجود دارد. این نوع از نظارت

داده ها بسیار مهم هستند به طوری که اشتباهات ممکن را شناسایی می کند یا حتی به قرار گرفتن مقادیری که دقیق هستند کمک

میکند یا حد اقل آسوده تریم به واسطه نادیده گرفتن آن ها در تحلیل های بعدی. دومین گراف نقطه Q-Q را نشان می دهد که با

موقعیت qq.plot به دست آمده است و مقادیر متغیر ها را در برابر چارک های تئوری توزیع نرمال قرار می دهد. موقعیت

همچنین پوششی با 95% فاصله مطمئن قرار می دهد. هما نظور که می بینیم بسیاری از مفادیر کم متغیر ها که به سادگی فرضیه

توزیع نرمال را رد می کنند با 95% اطمینان وجود دارد.

شما همچنین باید به استفاده گسترده ترکیب موقعیت در مثال گذشته با بسیاری موقعیت ها که نتایج سایرین هستند توجه کنید. شما همیشه

در درک این نوع از ساختار مشکل خواهید داشت، می توانید آن ها جداگانه در یک زمان برای شناخت کامل نام ببرید.

مثال دیگر (تصویر 2.3) این نوع از داده ها را که با استفاده هر ساختار زیر به دست می آید را با متغیر op04 نشان می دهد:

```

> boxplot(algae$oP04, ylab = "Orthophosphate (oP04)")
> rug(jitter(algae$oP04), side = 2)
> abline(h = mean(algae$oP04, na.rm = T), lty = 2)

```

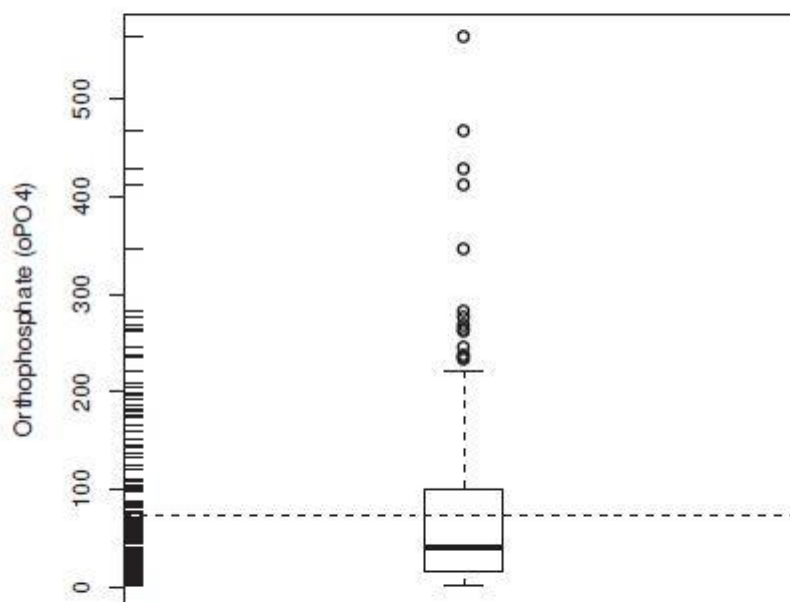


FIGURE 2.3: An “enriched” box plot for *orthophosphate*.

اولین ساختار جعبه نقاط متغیر 004 را طراحی کرده است. جعبه نقاط خلاصه ای سریع از برخی خواص کلیدی توزیع متغیر

فراهم می کند. برای مثال جعبه ای وجود دارد که محدودیت های افقی آن اولین و سومین چارک است. این جعبه دارای خطی

عمودی است در داخل که میانگین ارزش متغیر را نشان می دهد. اگر r چارک داخلی باشد. خط مورب کوچک عمودی بالای

جعبه بزرگترین مشاهده است که کم تر یا برابر با سومین چارک است. خط مورب عمودی زیر جعبه کوچک ترین مشاهده

است که بزرگتر یا برابر با اولین چارک است. دایره های زیر و بالای این خط های کوچک بیانگر مشاهداتی است که در مقایسه با

سایرین یا بسیار بالا است یا بسیار پایین. این بدین معنا است که جعبه نقاط مقدار زیادی اطلاعات به ما بدون توجه به ارزش

بنیادی و گسترده متغیر ها می دهد.

دومین ساختار از قبل قرار گرفته است (تنها تفاوت محل قرار گرفتن است). در حالی که `ablnhne()` برای کشیدن خطی

افقی با مقدار متغیر که با استفاده از `mean()` به دست آمده استفاده می کند. با مقایسه این خط با خط داخل جعبه که میانگین را

نشان می دهد به این نتیجه می رسیم که حضور مقدار زیادی خطوط ارزش میانگین را به عنوان آماری مرکزی از شکل انداخته

است. تحلیل تصویر 2.3 نشان می دهد که متغیر با توجه به مقادیر پایین توزیعی از مشاهدات را دارد است و همراه با انحرافی مثبت.

در بیشتر نمونه های آبی متغیر پایین است اما مشاهداتی فراوانی وجود دارد که نشان دهنده مقادیر بالاست.

برخی اوقات مقتی به خطوط برمیخوریم، نسبت به بازرسی مشاهداتی که این مقادیر عجیب را دارا هستند علاقه مند هستیم. ما دو راه

از این ها را نشلن خواهیم داد. ابتدا به صورت گرافیکی نشان می دهیم. اگر مقادیر متغیر `NH4` را به این شکل تعیین نماییم، به ارزشی

بسیار می‌رسیم. می‌توانیم نمونه آب را به این شکل شناسایی کنیم:

```
> plot(algae$NH4, xlab = "")
> abline(h = mean(algae$NH4, na.rm = T), lty = 1)
> abline(h = mean(algae$NH4, na.rm = T) + sd(algae$NH4, na.rm = T),
+       lty = 2)
> abline(h = median(algae$NH4, na.rm = T), lty = 3)
> identify(algae$NH4)
```

هولین ساختار تمامی مقادیر متغیر را تعیین میکند. موقعیت `abline()` سه نمونه خط را طراحی میکند، یکی با ارزشی متوسط، دیگری

را با معنی همراه با تقسیم استاندارد و دیگری را با میانگین. این‌ها برای این تکلیف شناسایی ضروری نیستند. آخرین ساختار بر یکدیگر

موثرند و به ما اجازه می‌دهد که بر نقاط معین چپ کلیک کنیم. برای هر نقطه کلیک شده R شواری از ستون عوامل را بر روی

صفحه اطلاعات می‌نویسد. ما می‌توانیم با راست کلیک کرد به فرآیند پایان دهیم. اگر می‌خواهیم که مشاهدات اسمی را بازرسی کنیم

بهتر است از پردازش زیر استفاده کنیم:

```
> plot(algae$NH4, xlab = "")
> clicked.lines <- identify(algae$NH4)
> algae[clicked.lines, ]
```

همان طور که قبلا حدس زدید (identify) به عنوان نتیجه به مات شواری از خطوط را که در همکاری با نقاط کلیک شده بودند

میدهد و بنا براین ما از این موضوع در صفحه اطلاعات algae استفاده می کنیم. می توانیم همچنین این بازرسی را بدون شکل اجرا

کنیم. همانند مقابل:

```
> algae[algae$NH4 > 19000, ]
```

این ساختار با اسفاده از عبارات منطقی شکل دیگری از صفحه اطلاعات را به تصویر می کشد. برون داد این ساختار قدری عجیب

به نظر می رسد. چنین نتیجه می دهد که تعدادی مشاهده با NA مقدار در نتغیر NH4 وجود دارد. در باره این مشاهدات R در ساخت

نتایج آن ها ناتوان است. می توانیم از این رفتار با استفاده از ساختار جلوگیری

کنیم. is.na() برداری را تولید می کند (درست یا غلط). عنصر بردار هنگامیکه NA، NH4 است درست است و این بردار دارای

عناصر بسیاری است که در صفحه اطلاعات موجود است. ساختار به وقتی باز می گردد که بردار با توجه به

محل نامشخص متغیر صحیح است. به طور خلاصه این جایگزین به ما مجموعه ای از داده ها را که دارای مقادیر مشخص هستند و

بیشتر از 19000 هستند می دهد. بیایید مثالی از نوع دیگری از بازرسی داده ها بزنیم. این مثال ها از پکیج R استفاده می کند. مجموعه

بزرگی از ابزار شکل های تکوین ایده را فراهم می کند.

تصور کنید که ما می خواهیم توزیع مقادیر خزه را بررسی کنیم. می توانیم از تمامی احتمالات پیش گفته استفاده کنیم. اما اگر بخواهیم

بینیم که آیا توزیع به سایر متغیر ها بستگی دارد یا نه بای از ابزار جدید استفاده کنیم.

نقاط معین ارائه های تصویری هستند که به عنصری اصلی بستگی دارد. عنصر متغیری است اسمی همراه با مجموعه ای از ارزش

های تعریف شده. برای مثال می توانیم به مجموعه ای از جعبه ای از نقاط برسیم. هر یک از نقاط با استفاده از نمونه های آب ها

به دست آمده است. این گراف ها نشان می دهد که این متغیر های اسمی چگونه بر توزیع مقادیر تاثیر دارند. کد رسیدن به جعبه نقاط

چنین است

```
> library(lattice)
> bwplot(size ~ a1, data=algae, ylab='River Size', xlab='Algal A1')
```

اولین ساختار در پکیج `lattice` ذخیره می شود. دومی از راه جعبه ای از نقاط به دست می آید. اولین منازعه در این باره چنین خوانده

می شود "نقطه `a1` برای مقدار سائز". منازعات باقی مانده دارای مغای می باشند.

تصویر 2.4 به ما اجازه می دهد که فراوانی های بالاتر را مشاهده کنیم که می تواند دانش موثری باشد.

یک نمونه مغایر با این نوع از نقاط که به درباره توزیع متغیر های معین اطلاعات می دهد نقاط جعبه درصدی هستند که در پکیج

`Hmisc` یافت می شود. بیاید مثالی از کاربرد آن با نقطه ای مشابه بینیم:

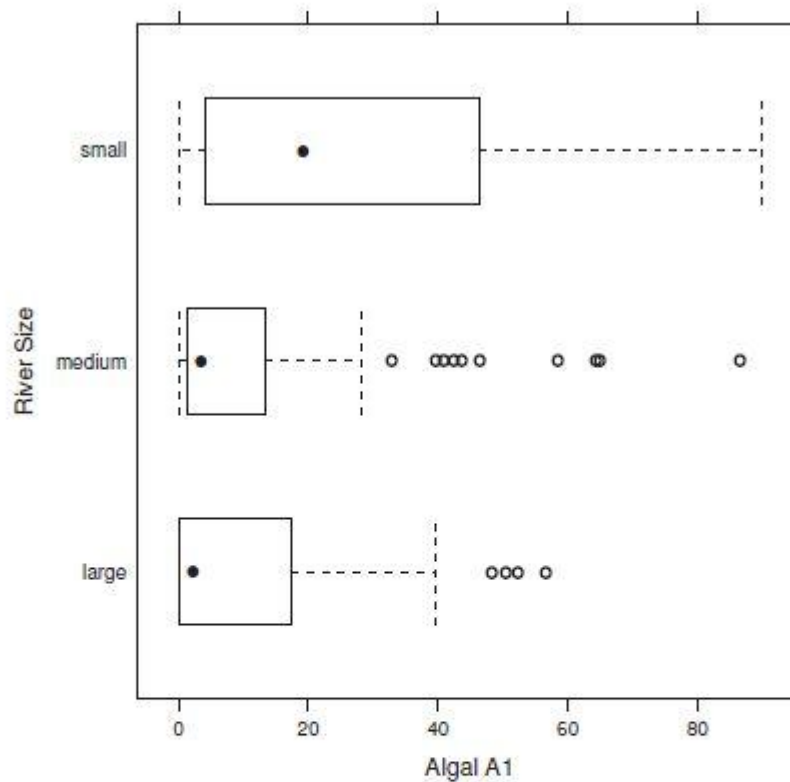


FIGURE 2.4: A conditioned box plot of Algal *a1*.

```
> library(Hmisc)
> bwplot(size ~ a1, data=algae, panel=panel.bpplot,
+        probs=seq(.01, .49, by=.01), datadensity=TRUE,
+        ylab='River Size', xlab='Algal A1')
```

نتیجه در شکل 2.5 نشان داده شده است. نقاط لبازر ارزشی فراوانی در سلیزهای متفاوت هستند. تصویر نشان می دهد که ارزش های

واقعی داده ها با خطوط مورب و اطلاعات توزیع این ارزش ها با نقاط چارک تهیه شده است. این نوع از نقاط تعیین شده به متغیر

اسمی و عنصر یگانه مر بوط نمی شود. بیایید مثالی را با توجه به مشاهده دفتار فراوان معین ببینیم، آخری متغیری اداام دار است.

تصویر 2.6 این گراف و کد را برای رسیدن به به نمونه زیر نشان می دهد:

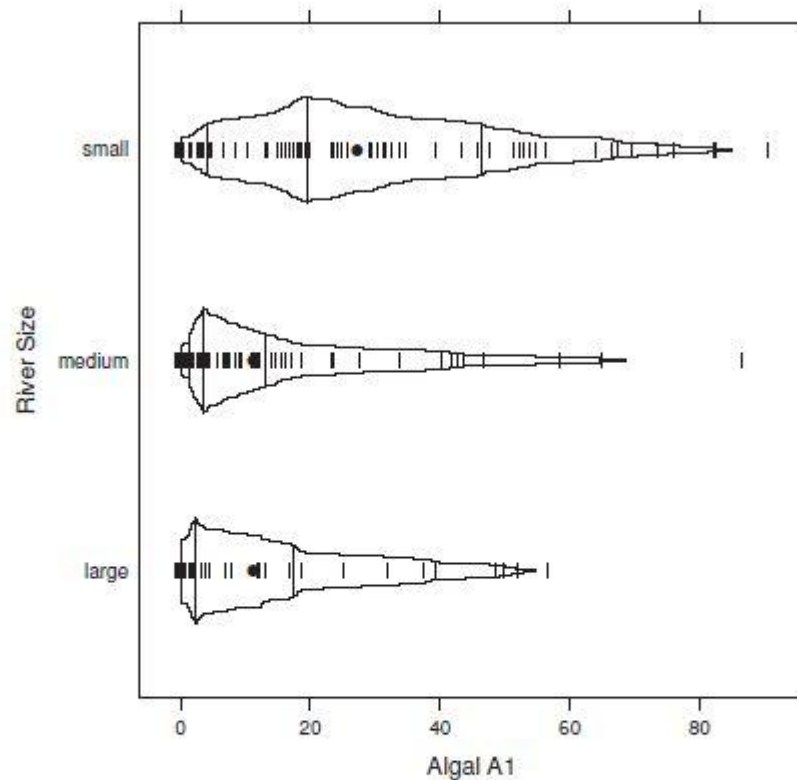


FIGURE 2.5: A conditioned box percentile plot of Algal *a1*.

```
> stripplot(season ~ a3/min02,  
+           data=algae[!is.na(algae$mn02),])
```

اولین ساختار به موجب ساخت یک سری سدازنده متغیر mno2 است. تغارها برای شامل شدن شواری برابر مشاهدات ساخته شده

اند. دلیل آن وجود مقادیر NA در متغیر است. ما از موقعیت na.omit() که هر گونه na را حذف می کند استفاده کرده ایم. دومین خط

شامل موقعیت گرافیکی (stripplot) می شود. این گرافی را می سازد که شامل ارزش های واقعی متغیر می شود در محل های

متفاوت با توجه به سایر متغیرها. تغارها از چپ به راست و از پایین به بالا چیده شده اند. وجود مقادیر NA در mno2 تاثیراتی بر

داده های استفاده شده برای کشیدن تصویر داشته است. به جای استفاده از پارامتر ما باید ستون های نمونه ها را با مقدار NA در

mno2 حذف کنیم.

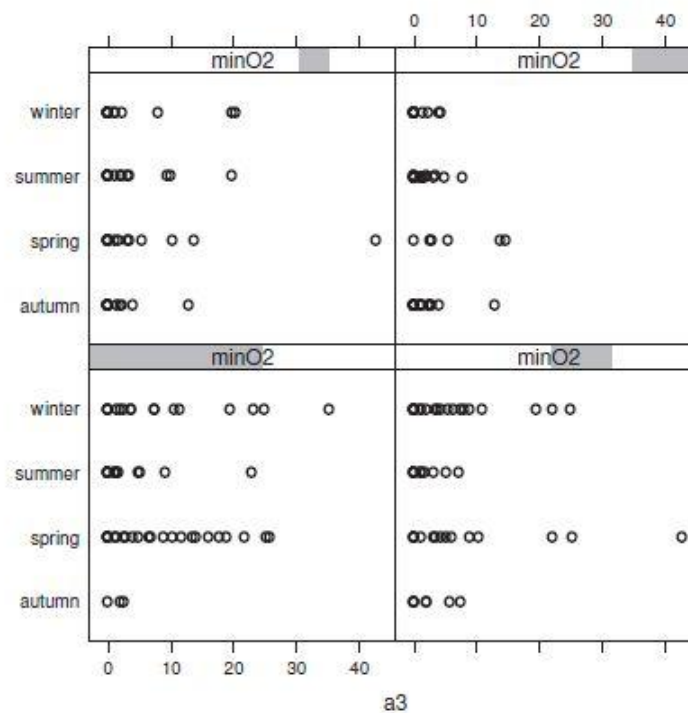


FIGURE 2.6: A conditioned strip plot of Algal $a3$ using a continuous variable.

parameter `data=algae` (as for creating Figure 2.4), we had to “eliminate” the rows corresponding to samples with NA values in `mnO2`.

مطالعات بیشتر بر تصویر ساز و خلاصه سازی داده ها

بیشتر کتاب های آماری استاندارد شامل خلاصه داده ها می شود. این کتاب مثال های ساده ای دارد و غیر رسمی است. کتاب تصویر

سازی داده ها نوشته کلواند یک اجبار است. کتاب رسمی تر در این زمینه کتاب عناصر داده های گرافیکی است. در نهایت کتاب

گرافیک های R نوشته مورل بسیار ریشه ای است.

هر گاه مجموعه از داده ها را در دست داشتیم می توانیم از راه کتر های زیر استفاده کنیم:

- موارد را با ناشناخته ها حذف کنید
- مقادیر ناشناخته را با کشف ارتباط بین متغیر ها پر کنید
- مقادیر ناشناخته را با کشف تشابه بین متغیر ها پر کنید
- از ابزاری که قادر به اندازه گیری مقادیر هستند استفاده کنید

جایگزین آخری بسیار مهم است چرا که مجموعه ابزار ها را محدود می کند. در بخش زیر نشان خواهیم داد که چگونه این راه کار ها

را در R تامین می کنیم. این به این معناست که اگر شما از روش دیگری استفاده کنید برای مقابله با مقادیر گم شده باید برای داشتن همه

موارد داده ها اصلی را بخوانید. آسان ترین راه برای انجام این کار استفاده از روش زیر است:

```
> library(DMwR)
> data(algae)
```

2.5.1 حذف مشاهدات با مقادیر ناشناخته

گزینه حذف موارد با مقادیر ناشناخته برای تامین بسیار آسان است و همچنین می تواند انتخابی معقول باشد برای نسبت ها با توجه به سبک داده های موجود. قل از به کارگیری تمامی مشاهدات با حداقل مقدار ناشناخته در برخی متغیرها عاقلانه است که نگاهی به آن داشته باشیم:

```
> algae[!complete.cases(algae),]
...
...
> nrow(algae[!complete.cases(algae),])
[1] 16
```

این موقعیت بردری را با بسیاری عناصر که در ستون ها هستند ایجاد می کند. بنابراین ساختار بالا نمونه های آبی را بابرخی مقادیر

Na نشان می دهد. برای حذف این 16 نمونه آبی می توانیم چنین کنیم:

```
> algae <- na.omit(algae)
```

حتی اگر نخواستیم از این روش حذف موارد استفاده کنیم می توانیم برخی از مشاهدات را به خاطر میزان بالای مقادیر ناشناخته

و بی استفاده حذف کنیم. با نگاه کردن به موارد ناشناخته می بینیم که دو مورد 62 و 199 دارای 6 از 11 مورد متغیر هستند. در این

موارد عاقلانه است نسبت به مشاهدات بی اعتنا باشیم:

```
> algae <- algae[-c(62, 199), ]
```

این ساختار به مجموعه ای از موقعیت ها ی R تعلق دارد.

در مورد این موقعیت می توانیم از هر نوعی در جهت یکی از موارد استفاده کنیم.ای امر موقت است زیرا تنها در این موقعیت قرار

میگیرد.موقعیت موقت ارقام سایر اشکال را محاسبه می کندبر اسا این کد می توانیم موقعیتی را بسزیم که دارای ارقام ناشناخته است.

این موقعیت در کتاب موجود است و شما می توانید به این شگل استفاده کنید:

```
> apply(algae, 1, function(x) sum(is.na(x)))
```

این موقعیت تنها در صورتی ضروری است که قبلا ستون هایی را حذف کرده باشید.در دومین منازعه می توانید از ارقام دیگری

استفاده کنید.بنابراین کدی است که برای دانستن ارقام ستون ها به کار می رود:

```
> data(algae)
> manyNAs(algae, 0.2)
```

```
[1] 62 199
```

در این مورد از مقدار دومین منازعه که 0.2 است استفاده کرده ایم

2.5.2 پر کردن ناشناخته ها با بیشترین مقدار فراوانی

جایگزینی برای استفاده از مقادیر ناشناخته این است که ارزش احتمالی هر یک از ناشناخته ها را بیابیم. راحت ترین و سریع تری راه

برای پر کردن مقادیر ناشناخته استفاده از برخی اصول آماری است. بسیاری از اصول آماری مانند ابزار میانگین، مد و... هستند. برای

توزیع برابر در جایی که همه مشاهدات وجود دارند این آمار بهترین روش است. به بیانی دیگر وجود خطوط ممکن است محاسبات را

بد شکل جلوه دهد. بنابراین استفاده از ابزار بدون بازرسی قبلی درست نیست. برای توزیع منحرف یا برای متغیر ها میانگین باید

آماری باشد. برای مثال نمونه [48]algea در متغیر mxPH مقداری ندارد زیرا توزیع این متغیر نرمال است و ما میتوانیم از میانه آن

برای پر کردن جای خالی استفاده کنیم. به این شکل انجام می شود:

موقعیت میانه به این معناست که ارزش هر یک از این بردار ها ارزش بردار دیگری را در محاسبات نشان می دهد. بیشتر وقت ها

مه به پ کردن ستون به جای کار کردن روی نوارد علاقه مند هستیم. این متغیر روی 12 نمونه آب ناشناخته است. توزیع Chla به

مقداری پایین تر رسده و مقدار کمتری وجود دارد که میانه را مشخص می کند. بنابراین از میانگین برای پر کردن ستون های ناشناخته

استفاده می کنیم.

```
> algae[48, "mxPH"] <- mean(algae$mxPH, na.rm = T)
```

این موقعیت که در کتاب موجود است همه ناشناخته ها را پر می کند. این موقعیت از میانگین برای ستون ها و بیشترین مقدار

فراوانی استفاده می کند. از این می توانید استفاده کنید:

```
> data(algae)
> algae <- algae[-manyNAs(algae), ]
> algae <- centralImputation(algae)
```

اما حضور مقادیر ناشناخته استفاده از ای روش ها را ناممکن ساخته. این راه کار ساده با وجود سرعت برای داده های بزرگ قابل

استفاده است و بر تحلیل آینده ما تاثیر خواهد داشت. اگرچه روش های بی پایه برای ناشناخته ها پیچیده هستند و ممکن است برای داده

های بزرگتر مشکل ایجاد کنند.

2.5.3 پر کردن ناشناخته ها با کشف ارتباط ها

یکی از راه های کشف ارتباطات ناشناخته رسیدن به ارتباط بین متغیر ها است. این روش بر استفاده از میانه مرجح است برای رسیدن

به ارتباط متغیر ها به امر زیر اشاره می کنیم:

```
> cor(algae[, 4:18], use = "complete.obs")
```

این امر مانریسی را با ارتباط بین متغیر ها ایجا می کند. موقعیت دیگر R می تواند در شکل ساختاری این خط ارتباطی مورد استفاده

قرار بگیرد که به ما اجازه می دهد که مقادیر متغیر ها را حدس بزنیم. نتیجه این ساختار خیلی محسوس نیست اما می توانیم با استفاده

از آن موقعیت دیگری را توسعه دهیم.

```

> symnum(cor(algae[,4:18],use="complete.obs"))

      mP m0 C1 N0 NH o P Ch a1 a2 a3 a4 a5 a6 a7
mxPH 1
mn02   1
C1     1
N0     1
NH     , 1
oP04   . . 1
P04    . . * 1
Chla . . . 1
a1     . . . 1
a2     . . . 1
a3     . . 1
a4     . . 1
a5     . . 1
a6     . . 1
a7     . . 1
attr(,"legend")
[1] 0 ' ' 0.3 '.' 0.6 ', ' 0.8 '+' 0.9 '*' 0.95 'B' 1

```

این ارائه سمبلی برای ماتریس های ارتباطی محسوس تر است. در داده ها ما از تباط ها ب اهمیت هستند. این دو متغیر آخر بسیار با

هم در ارتباط هستند و بنابراین استفاده از آن برا پر کردن ناشناخته ها ریسک است. با توجه به PO4, oPO4 کشف این ارتباط به ما

اجازه می دهد که این متغیر عا را پر کنیم برای رسیدن به این مورد باید ارتباط بین این متغیر ها را کشف کنیم:

```

> data(algae)
> algae <- algae[-manyNAs(algae), ]
> lm(P04 ~ oP04, data = algae)

Call:
lm(formula = P04 ~ oP04, data = algae)

Coefficients:
(Intercept)          oP04
      42.897         1.293

```

مورد استفاده قرار بگیرد.مدل

این موقعیت می تواند برای رسیدن به مدل های خطی
خطی ک به دست

با این فرمولی توانیم ناشناخته ها را پر کنیم تگر که

آورده ایم به ما می گوید که
همه آن ها ناشناخته

نباشند.

قبل از حذف نمونه های 62 و 199 با مشاهده ای رو به رو میشویم.پس به سادگی می توانیم از این رابطه
استفاده کنیم:


```
> algae[28, "P04"] <- 42.897 + 1.293 * algae[28, "oP04"]
```

اگرچه برای اهداف تصویری باید حدس زد بسیاری نمونه با مقادیر ناشناخته وجود دارد. بهتری کار آن است که ساختاری برای مقدار

P04 و سپس اسن ساختار را:

```
> data(algae)
> algae <- algae[-manyNAs(algae), ]
> fillP04 <- function(oP) {
+   if (is.na(oP))
+     return(NA)
+   else return(42.897 + 1.293 * oP)
+ }
> algae[is.na(algae$P04), "P04"] <- sapply(algae[is.na(algae$P04),
+   "oP04"], fillP04)
```

اول ساختاری با این نام می سازیم. این ساختار برای تمامی نمونه ها مورد استفاده است. این ساختار دارای برداری است و نتیجه اش

برداری دیگر است با همان طول و همان عناصر و هر عنصر برای برداری مفروض است. این بدین معناست که لین امر دارای

نتیجه برداری است. آخرین فرضیه به صورت مثالی است از استفاده ترکیب ساختاری. مطالعه ارتباط خطی ما را قادر می سازد که

برخی از ناشناخته های جدید را پر کنیم.می توانیم از هیستوگرام های معین که در پکیج موجود است برای لین مشکل استفاده کنیم.

برای مثال در شکل 2.7 نمونه ای از گراف را مشاهده می کنیم.این گراف به این شکل ایجاد شده است:

```
> histogram(~mxPH | season, data = algae)
```

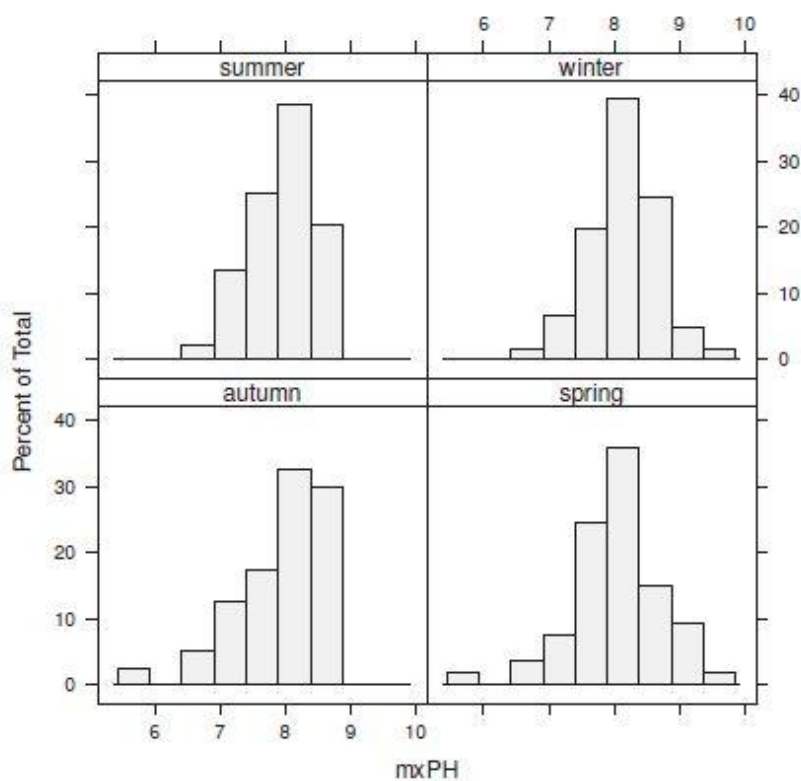


FIGURE 2.7: A histogram of variable *mxPH* conditioned by *season*.

این ساختار هیستوگرامی را در مقادیری برای ارزش های متفاوت به دست می آورد. ممکن است به این امر دقت کرده باشید نظم

فصول در گراف ها کمی غیر طبیعی است. اگر خواهان نظم طبیعی هستید باید نوع چیدمان را در صفحه اطلاعات تغییر دهید.

به این شکل انجام می شود:

```
> algae$season <- factor(algae$season, levels = c("spring",  
+ "summer", "autumn", "winter"))
```

هنگامی که به مجموعه از ارزش های متغیر ها بر می خوریم سطح پارامتر هایی را که به نظم هستند را حدس خواهیم زد به

هیستوگرام ها در تصویر 2.7 دقت کنید که شبیه هم هستند و ما به سمت این نتیجه گیری که مقادیر تحت تاثیر فصول نیستند سوق

می دهد. اگر سعی به استفاده از همان رودخانه را داشته باشیم مشاهده می کنیم که رودخانه های کوچک تر دارای مقادیر کم تری

از متغیر ها هستند. می توانیم مطالعات خود را با استفاده از متغیر های اسمی گسترش دهیم. برای مثال

```
> histogram(~mxPH | size * speed, data = algae)
```

متغیری ثابت را برای تمامی رودخانه ها نشان می دهد. تنها نمونه ای که این خصوصیت را دارد نمونه 48 است. سایر جایگزین ها

به این اطلاعات مشابه رسیده اند اما اکنون مقادیر دیگری از این متغیر قابل توجه است

```
> stripplot(size ~ mxPH | speed, data = algae, jitter = T)
```

نتیجه این ساختار در تصویر 2.8 نشان داده شده است. این نوع تحلیل برای سایر متغیر ها هم انجام میگیرد و هنوز هم فرآیندی موثر

است زیرا بسیاری ترکیب برای تحلیل وجود دارد. این روشی است که در مورد داده های کوچک با متغیر های کم مورد استفاده قرار

می گیرد.

2.5.4 پر کردن مقادیر ناشناخته به وسیله کشف موارد مشابه

به جای کشف ارتباط بین متغیر ها می توانیم تشابهات بین مشاهدات را کشف کنیم. برای دوباره به کار انداختن کد قبلی بهتر است که داده ها را دوباره بخوانیم.

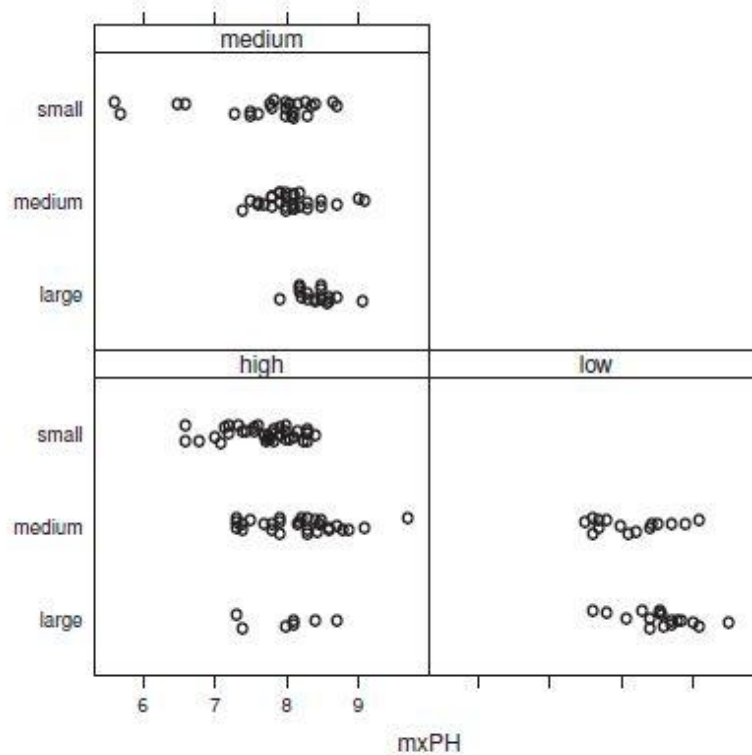


FIGURE 2.8: The values of variable *mxPH* by river size and speed.

```
> data(algae)
> algae <- algae[-manyNAs(algae), ]
```

این نظریه چنین بیان می کند که اگر دو نمونه آب مشابه باشند و یکی از آن ها مقدار نامشخص داشته باشد احتمال می رود که مقدارشان

مثل هم باشد. این نکته به واسطه استفاده از فضای چندگانه متری متغیر ها برای توصیف مشاهدات تعریف شده است این فاصله می تواند

به عنوان مربعی از تفاوت های مربعی در دو مورد تعریف شود که به این صورت است

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2} \quad (2.1)$$

روشی که ما در زیر توصیف کرده ایم از این متریک برای یافتن ده مورد از این موارد د نمونه های آ ب مورد استفاده قرار می

گیرد. ما در مورد دو راه استفاده آن ها بحث می کنیم. اولی میانگین را از بین ده تا از نزدیک ترین همسایه ها اندازه می گیرد و ما

از بیشترین فراوانی بین همسایه ها استفاده می کنیم. دومین روش از میانگین همسایه ها استفاده می کند. از موقعیتی استفاده می کنیم

برای رسیدن به اندازه فاصله ها.

$$w(d) = e^{-d} \quad (2.2)$$

این نظریه در موقعیت دیگری تشریح شده است. این متغیر به عملکرد اجازه می دهد که با هر دو متغیر ها مجموعه ای را تشکیل

دهد. که به شکل زیر است:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^p \delta_i(\mathbf{x}_i, \mathbf{y}_i)} \quad (2.3)$$

در جایی که قرار دارد فاصله بین دو مقدار متغیر i به این شکل مفروض است

$$\delta_i(v_1, v_2) = \begin{cases} 1 & \text{if } i \text{ is nominal and } v_1 \neq v_2 \\ 0 & \text{if } i \text{ is nominal and } v_1 = v_2 \\ (v_1 - v_2)^2 & \text{if } i \text{ is numeric} \end{cases} \quad (2.4)$$

فاصله ها پس از نرمال شدن به این شکل محاسبه می شود

$$y_i = \frac{x_i - \bar{x}}{\sigma_x} \quad (2.5)$$

اکنون باید دید که چگونه باید از این موقعیت استفاده کنیم:

```
> algae <- knnImputation(algae, k = 10)
```

اگر می خواهید که از راه کار استفاده از میانگین استفاده کنید از این روش استفاده کنید

```
> algae <- knnImputation(algae, k = 10, meth = "median")
```

برای تصمیم گیری درباره استفاده از کدام روش همواره پاسخی دقیق وجود دارد. روش دوم با وجود مشکلاتی منطقی تر به نظر

میرسد. برای این مشکلات بزرگ می توانیم از نمونه های تصدفی برای محاسبه تشابهات استفاده کنیم.

2.6 به دست آوردن مدل های پیش بینی شده

هدف از این مطالعات رسیدن به پیش بینی های مقادیر فراوانی های 140 نمونه است. این موضوع شامل تلاش برای رسیدن به مدلی

است که با متغیر ها در ارتباط است. در این بخش ابتدا به دو تفاوت مدل ها اشاره می کنیم: برگشت ضرب خطی و شعبه های برگشت

این مدل ها دو نمونه خوب هستند برای مشکلات برگشتی که آن ها با توجه به شکل آن ها متفاوت هستند. این به این معنا نیست که ما

نباید از این داده ها استفاده کنیم و نباید از بین آن ها 140 نمونه را انتخاب کنیم. مدل هایی که ما در حال بررسی هستیم مقادیر متفاوتی

را تامین می کند. همینطور راه دیگری را در بخش 2.5 برای پیش پردازش در پیش می گیریم. با توجه به شعبه های برگشت از 200

نمونه اصلی استفاده می کنیم. در تحلیل هایی که انجام می دهیم گمان بر این است که مقدار اصلی را نمیدانیم. این مقادیر تنها برای شما

مفروض است تا به نظریه نهایی برسید.

برگشت ضرب خطی از تکنیک های مورد استفاده در آمار است. همان طور که قبلا گفته شده راه معینی برای تامین مقادیر وجود

ندارد. اگر چه قبلا از این روش استفاده کرده ایم نمونه های 62 و 199 را حذف می کنیم. کد مقابل به صفحه اطلاعاتی بدون مقدار گم شده می رسد:

```
> data(algae)
> algae <- algae[-manyNAs(algae), ]
> clean.algae <- knnImputation(algae, k = 10)
```

پس از استفاده از این کد داده هایی داریم که بدون مقادیر مفقود هستند. بیا با یاد گرفتن چگونگی رسیدن به برگشت خطی آغاز کنیم.

```
> lm.a1 <- lm(a1 ~ ., data = clean.algae[, 1:12])
```

این ساختار به مدل برگشت خطی اشاره دارد. در این مثال بیان می کند که ما مدلی را متغیر ثابت پیش بینی کرده ایم برای مثال اگر

مدلی را دو متغیر خاص پیش بینی کنیم باید چنین مدلی را هم بیان کنیم. متغیرهای دیگری هم وجود دارد که ما به

عنوان ضرورت معرفی می کنیم. نتیجه این ساختار شکلی است که شامل مدل خطی می شود. به جزئیات بیشتری می رسم اگر ساختار

پایین را دنبال کنیم:

```
> summary(lm.a1)
```

```
Call:
```

```
lm(formula = a1 ~ ., data = clean.algae[, 1:12])
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-37.679	-11.893	-2.567	7.410	62.190

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	42.942055	24.010879	1.788	0.07537 .
seasonspring	3.726978	4.137741	0.901	0.36892
seasonsummer	0.747597	4.020711	0.186	0.85270
seasonwinter	3.692955	3.865391	0.955	0.34065
sizemedium	3.263728	3.802051	0.858	0.39179
sizesmall	9.682140	4.179971	2.316	0.02166 *
speedlow	3.922084	4.706315	0.833	0.40573
speedmedium	0.246764	3.241874	0.076	0.93941
mxPH	-3.589118	2.703528	-1.328	0.18598
mnO2	1.052636	0.705018	1.493	0.13715
Cl	-0.040172	0.033661	-1.193	0.23426
NO3	-1.511235	0.551339	-2.741	0.00674 **
NH4	0.001634	0.001003	1.628	0.10516
oP04	-0.005435	0.039884	-0.136	0.89177
P04	-0.052241	0.030755	-1.699	0.09109 .
Chla	-0.088022	0.079998	-1.100	0.27265

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 17.65 on 182 degrees of freedom
```

```
Multiple R-squared: 0.3731, Adjusted R-squared: 0.3215
```

```
F-statistic: 7.223 on 15 and 182 DF, p-value: 2.444e-12
```

قبل از آنت که اطلاعات را تحلیل کنیم درباره کنترل متغیر ها چیز هایی می گوییم. به طور مثال برای هر عامل R عبرت $k-1$ را می

سازد. مقدار 1 به این معنا است که مقدار عامل حاضر است و و به این معنا است که مقدار متغیر 0 است. با نگاه کردن به خلاصه در

بالا می بینیم که R سه نوع متغیر اصلی برای عامل فصل ساخته است. کاربرد ساختار غصول در مدل خطی برخی اطلاعات در گیر

با مدل ها را به ما می دهد. این ها باید میانه 0 و توزیعی عهادلانه داشته باشند. برای کنترل اهمیت هر یک از آنها می توانیم فرضیه

ها را تست کنیم که به این شکل است. برای تست کردن این فرضیه ها روش تی تست استفاده می شود. مقدار 0.0001

به این معنا است که ما 99.99% مطمئنیم که این عوامل پوچ است. به طور خلاصه تنها برای عواملی که دارا نمونه هستند می توانیم

فرضیه را رد کنیم. این نشان می دهد که درجه تطابق مدل با داده ها به واسطه مدل بیا ن شده است. عوامل تنظیم شده بیشتر مورد

تقاضا بوده تا پارامتر های مدل برگشتی. در نهایت می توانیم فرضیه پوچ را هم که دارای هیچ وابستگی به متغیر ها نیستند هم تست

کنیم که به این شکل هستند

این آمار برای این هدف ها مورد استفاده قرار می گیرد و

سطح 0.0001

به این معنا است که ما مطمئنیم که فرضیه صحیح نیست. برخی اطلاعات دیگر هم ممکن است چک شوند یکی از راه ها به سادگی هر

یک از مقصد ها را در مقابل متغیر دیگر قرار می دهد. اشتباهات بزرگتر به واسطه اضافه کردن ستون ها علامت می خورد بنابراین

میتوان مشاهدات را بازرسی کرد. نسبت متغیرها که به واسطه این مدل توضیح داده شده موثر نیست. با توجه به برخی از عوامل

نتیجه را در این مدل سوال می کنیم. در این بخش متدی را با نام حذف عقبی مطالعه می کنیم. با استفاده از ساختار مدل خطی مطالعه

خود را آغاز می کنیم. لین کاهش هایی است در برخی مربع ها. نتیجه این تحلیل به شکل زیر است

```
> anova(lm.a1)
```

Analysis of Variance Table

Response: a1

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
season	3	85	28.2	0.0905	0.9651944	
size	2	11401	5700.7	18.3088	5.69e-08	***
speed	2	3934	1967.2	6.3179	0.0022244	**
mxPH	1	1329	1328.8	4.2677	0.0402613	*
mnO2	1	2287	2286.8	7.3444	0.0073705	**
Cl	1	4304	4304.3	13.8239	0.0002671	***
NO3	1	3418	3418.5	10.9789	0.0011118	**
NH4	1	404	403.6	1.2963	0.2563847	
oP04	1	4788	4788.0	15.3774	0.0001246	***
P04	1	1406	1405.6	4.5142	0.0349635	*
Chla	1	377	377.0	1.2107	0.2726544	
Residuals	182	56668	311.4			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

These results indicate that the variable *season* is the variable that least contributes to the reduction of the fitting error of the model. Let us remove it from the model:

```
> lm2.a1 <- update(lm.a1, . ~ . - season)
```

خلاصه اطلاعات به این شکل است:

```
> lm2.a1 <- update(lm.a1, . ~ . - season)
```

تقریبا توسعه یافته است اما هنوز خیلی موثر نیست. می توانیم مقایسه ای را بین دو مدل با استفاده از مدل ها
طراح کنیم و این بار با

این با این دو مدل:


```
> summary(lm2.a1)
```

Call:

```
lm(formula = a1 ~ size + speed + mxPH + mnO2 + Cl + NO3 + NH4 +  
    oP04 + P04 + Chla, data = clean.algae[, 1:12])
```

Residuals:

	Min	1Q	Median	3Q	Max
	-36.460	-11.953	-3.044	7.444	63.730

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	44.9532874	23.2378377	1.934	0.05458 .
size _{medium}	3.3092102	3.7825221	0.875	0.38278
size _{small}	10.2730961	4.1223163	2.492	0.01358 *
speed _{low}	3.0546270	4.6108069	0.662	0.50848
speed _{medium}	-0.2976867	3.1818585	-0.094	0.92556
mxPH	-3.2684281	2.6576592	-1.230	0.22033
mnO2	0.8011759	0.6589644	1.216	0.22561
Cl	-0.0381881	0.0333791	-1.144	0.25407
NO3	-1.5334300	0.5476550	-2.800	0.00565 **
NH4	0.0015777	0.0009951	1.586	0.11456
oP04	-0.0062392	0.0395086	-0.158	0.87469
P04	-0.0509543	0.0305189	-1.670	0.09669 .
Chla	-0.0841371	0.0794459	-1.059	0.29096

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.57 on 185 degrees of freedom

Multiple R-squared: 0.3682, Adjusted R-squared: 0.3272

F-statistic: 8.984 on 12 and 185 DF, p-value: 1.762e-13

این ساختار تحلیلی را درباره دو مدل با استفاده از اف تست برای دسترسی به اهمیت تفاوت ها انجام می دهد. باید به خاطر داشته

باشیم که این مدل جدید آسان تر است. این فرآیند تا وقتی که هیچ گونه عاملی نداشته باشیم ادامه دارد. برای آسان تر کردن فرایند

حذف کردن ساختاری داریم که این کار را انجام می دهد. کد مقابل مدلی خطی که از به کارگیری روش حذف نتیجه می شود را

می سازد:

```
> anova(lm.a1,lm2.a1)
```

Analysis of Variance Table

Model 1: a1 ~ season + size + speed + mxPH + mn02 + Cl + N03 + NH4 + oP04 + P04 + Chla

Model 2: a1 ~ size + speed + mxPH + mn02 + Cl + N03 + NH4 + oP04 + P04 + Chla

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
1	182	56668				
2	185	57116	-3	-448	0.4792	0.6971

این موقعیت از داده ها مرکزی برای اجرای مدل تحقیقاتی استفاده می کند. این تحقیقات از حذف عقبی همراه با پارامتر جهت یاب

استفاده می کند. می توانیم به اطلاعات در آخرین مدل دست بیابیم:

```
> final.lm <- step(lm.a1)
```

Start: AIC= 1151.85

```
a1 ~ season + size + speed + mxPH + mn02 + Cl + N03 + NH4 + oP04 +  
P04 + Chla
```

	Df	Sum of Sq	RSS	AIC
- season	3	425	57043	1147
- speed	2	270	56887	1149
- oP04	1	5	56623	1150
- Chla	1	401	57018	1151
- Cl	1	498	57115	1152
- mxPH	1	542	57159	1152
<none>			56617	1152
- mn02	1	650	57267	1152
- NH4	1	799	57417	1153
- P04	1	899	57516	1153
- size	2	1871	58488	1154
- N03	1	2286	58903	1158

Step: AIC= 1147.33

```
a1 ~ size + speed + mxPH + mn02 + Cl + N03 + NH4 + oP04 + P04 +  
Chla
```

	Df	Sum of Sq	RSS	AIC
- speed	2	213	57256	1144
- oP04	1	8	57050	1145
- Chla	1	378	57421	1147
- mn02	1	427	57470	1147
- mxPH	1	457	57500	1147
- Cl	1	464	57506	1147
<none>			57043	1147
- NH4	1	751	57794	1148
- P04	1	859	57902	1148
- size	2	2184	59227	1151
- N03	1	2353	59396	1153

نسبت این متغیر ها در این مدل خیلی هم جالب نیست.

2.6.2 شعبه های برگشت

بیا یید به مدل دیگری از این موارد نگاه کنیم. برای مثال یاد خواهیم گرفت که چگونه به این شعب دست یابیم این مدل ها داده های با

مقادیر کم را کنترل می کند و ما باید نمونه های 29 . 199 را حذف کنیم. ساختار لازم برای دست یابی به این شعب چنین بیان شده:

```
> library(rpart)
> data(algae)
> algae <- algae[-manyNAs(algae), ]
> rt.a1 <- rpart(a1 ~ ., data = algae[, 1:12])
```

اولین ساختار در پکیج ذخیره می شود. آخرین ساختار به شعبه می رسد. دومین منازعه بیان می دارد که کدام داده مورد نیاز است.

```

> rt.a1

n= 198

node), split, n, deviance, yval
* denotes terminal node

1) root 198 90401.290 16.996460
2) P04>=43.818 147 31279.120 8.979592
4) C1>=7.8065 140 21622.830 7.492857
8) oP04>=51.118 84 3441.149 3.846429 *
9) oP04< 51.118 56 15389.430 12.962500
18) mn02>=10.05 24 1248.673 6.716667 *
19) mn02< 10.05 32 12502.320 17.646870
38) N03>=3.1875 9 257.080 7.866667 *
39) N03< 3.1875 23 11047.500 21.473910
78) mn02< 8 13 2919.549 13.807690 *
79) mn02>=8 10 6370.704 31.440000 *

```

شعب برگشتی از تست های منطقی است. یک شعبه از رو ریشه علامت دار خوانده می شود. R. برخی از این اطلاعات را فراهم می

کند. برای مثال می توانیم 198 نمونه را در این مورد مشاهده کنیم که این نمونه ها دارای میانگینی با مقدار فراوانی 16.99 هستند و

هر یک از میانگین ها 90401.29 است. این ها به نتیجه آزمایش مربوط است. برای مثال از ریشه یک ساقه داریم برای موارد آزمایشی

در نمونه 147. از بند دوم دو شاخه دگر داریم که به بند های 4 و 5 می رسد. این بند ها با ستاره مشخص شده اند این به این معنا

است آگد بخواهیم از درختی استفاده کنیم تنها بهید از شاخه به ریشه آن را دنبال کنیم. میانگین مورد نظر متغیر در برگی که به آن

برسیم یافت می شود. همچنین می توانیم به ارائه گرافیکی درخت برسیم. این ساختار ها دارای پارامتر های زیادی هستند برای کنترل

تصویر سازی درخت. استفاده از آن در درخت به دست آمده به نتیجه موجود در تصویر 2.9 می رسیم.

این ساختار می تواند در اشکال درخت مورد استفاده قرار بگیرد. انشعابات آخر قسمتی از راه کار مورد استفاده در این مدل است.

درخت ها معمولا دارای دو مرحله هستند. این فرایند دارای هدفی است در جهت جلوگیری از تطابق بیش از حد. این موضع مربوط

می شود به این حقیقت یک درخت بزرگ داده ها را کاملاً پوشش ندهد. مشکل تطابق بیش از حد در بسیاری از مدل ها وجود دارد

این مدل ها با وجود استفاده گسترده دارای این مشکل است. این ساختار که ما برای رسیدن به درخت از آن استفاده کرده ایم در

برخورد با ملاک اصلی متوقف می شود. آستانه ها به واسطه پارامتر ها می کنترل می شود. مقدار آن ها 0.01، 0.20 و 30 است. اگر

بخواهیم از مشکل تطابق بیش از حد جلوگیری کنیم باید همیشه از کنترل ارزش این ملاک ها جاو گیری کنیم این پکیج روشی را

به نام ارزش پیچیدگی تکمیل می کند این روش شاخه زنی سعی دارد که مقداری را که بهترین مقایسه را بیان می کند حدس بزند.

این اطلاعات با استفاده از این ساختار به دست می آید:

IranDataMiner.ir

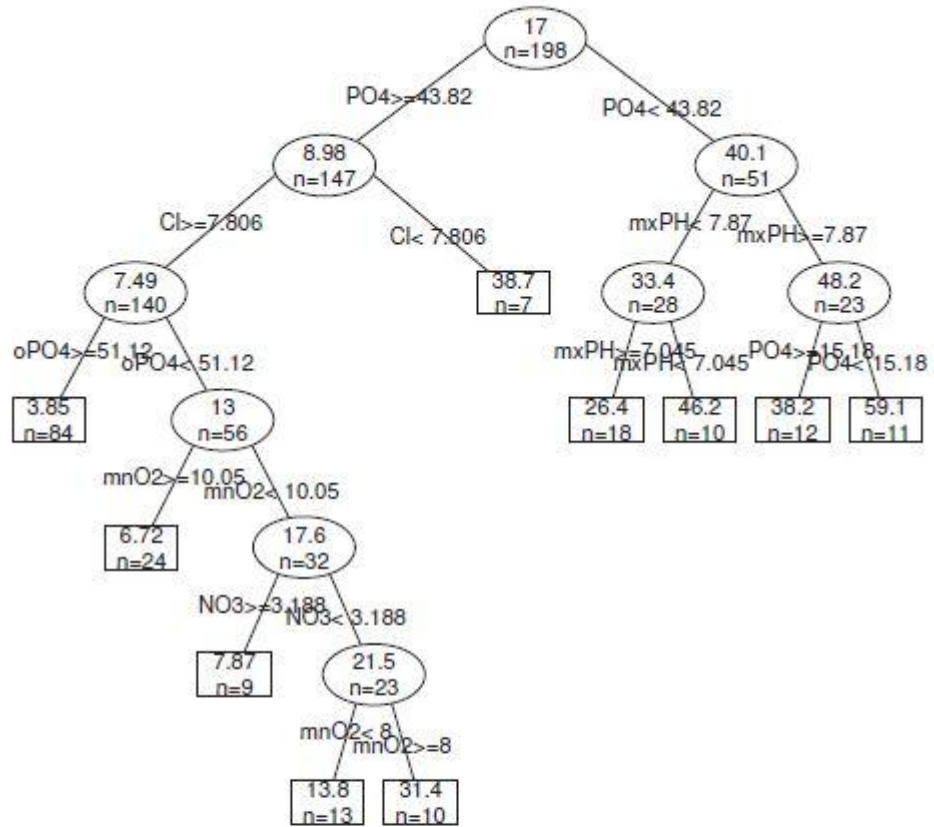


FIGURE 2.9: A regression tree for predicting algal *a1*.

این درخت با این ساختار ساخته شده. اگرچه R با استفاده از فرایند داخلی حدس می زند که این درخت دارای میانگینی است. قانون

انتخاب جایگزین این است که بهترین را بر اساس قوانین انتخاب کرد. در این مورد درخت بسیار کوچک است و دارای خطایی کمتر

از $0.78625 = 0.10892 + 0.67733$ است. اگر این درخت را به درخت پیشنهاد شده ترجیح بدهیم به این میرسیم:

```
> printcp(rt.a1)
```

Regression tree:

```
rpart(formula = a1 ~ ., data = algae[, 1:12])
```

Variables actually used in tree construction:

```
[1] C1 mn02 mxPH N03 oP04 P04
```

Root node error: 90401/198 = 456.57

n= 198

	CP	nsplit	rel error	xerror	xstd
1	0.405740	0	1.00000	1.00932	0.12986
2	0.071885	1	0.59426	0.73358	0.11884
3	0.030887	2	0.52237	0.71855	0.11518
4	0.030408	3	0.49149	0.70161	0.11585
5	0.027872	4	0.46108	0.70635	0.11403
6	0.027754	5	0.43321	0.69618	0.11438

در این کتاب این ساختار با فرایندی اتوماتیک وار تهیه کرده ایم:


```
> rt2.a1 <- prune(rt.a1, cp = 0.08)
> rt2.a1
```

```
n= 198
```

```
node), split, n, deviance, yval
* denotes terminal node
```

```
1) root 198 90401.29 16.996460
2) P04>=43.818 147 31279.12 8.979592 *
3) P04< 43.818 51 22442.76 40.103920 *
```

R نوعی از روش هرس درخت را در این ساختار معرفی می کند.اولی شامل شماره هایی از بند ها می شود که می خواهیم

آن را هرس کنیم:

```
> (rt.a1 <- rpartXse(a1 ~ ., data = algae[, 1:12]))
```

```
n= 198
```

```
node), split, n, deviance, yval
* denotes terminal node
```

```
1) root 198 90401.29 16.996460
2) P04>=43.818 147 31279.12 8.979592 *
3) P04< 43.818 51 22442.76 40.103920 *
```

توجه کنید که این ساختار به درختی بر می گردد که به این معنا است می توانید درخت هرس شده را ذخیره کنید

شما می توانید از این ساختار در شکلی گرافیکی استفاده کنید. اگر با موس روی بعضی بندها کلیک کنید اطلاعاتی در بند پرینت

گرفته می شود. اگر دوباره روی آن کلیک کنید درخت را روی بند هرس می کند. این فرایند را با راست کلیک به پایان می رسانید

نتیجه به این شکل است:

```
> first.tree <- rpart(a1 ~ ., data = algae[, 1:12])
> snip.rpart(first.tree, c(4, 7))
```

```
n= 198
```

```
node), split, n, deviance, yval
* denotes terminal node
```

```
1) root 198 90401.290 16.996460
 2) P04>=43.818 147 31279.120 8.979592
   4) C1>=7.8065 140 21622.830 7.492857 *
   5) C1< 7.8065 7 3157.769 38.714290 *
 3) P04< 43.818 51 22442.760 40.103920
   6) mxPH< 7.87 28 11452.770 33.450000
      12) mxPH>=7.045 18 5146.169 26.394440 *
      13) mxPH< 7.045 10 3797.645 46.150000 *
   7) mxPH>=7.87 23 8241.110 48.204350 *
```

2.7 انتخاب و بررسی مدل

در بخش 2.6 دو مثال در مورد پیش بینی مدل ها که در این مطالعات مورد استفاده قرار می گیرد می بینیم. برای پاسخ به سوال در

مورد نمونه ها باید به معین کردن مراجع و ملاک ها توجه کنیم که به این معناست به معین کردن چگونگی ارزیابی اجرای مدل ها

نیاز داریم بسیاری از ملاک ها برای ارزیابی مدل ها وجود دارد. سلیر ملاک ها مثل سایر مدل ها وجود دارد که می تواند برای هر یک

از مشکلات دلد ه مهم باشد. اجرای غیر مستقیم این مدل ها به واسطه مقایسه پیش بینی ها با مقادیر واقعی به دست آمده است. اولین

قدم برای دست یافتن به پیش بینی ها جایی است که می خواهیم آن را ارزیابی کنیم. این ساختار کلی به داده ها و مدل ها می رسد

```
> (mae.a1.lm <- mean(abs(lm.predictions.a1 - algae[, "a1"])))
```

```
[1] 13.10681
```

```
> (mae.a1.rt <- mean(abs(rt.predictions.a1 - algae[, "a1"])))
```

```
[1] 11.61717
```

Another popular error measure is the mean squared error (MSE). This measure can be obtained as follows:

```
> (mse.a1.lm <- mean((lm.predictions.a1 - algae[, "a1"])^2))
```

```
[1] 295.5407
```

```
> (mse.a1.rt <- mean((rt.predictions.a1 - algae[, "a1"])^2))
```

```
[1] 271.3226
```

ای دو عبارت پیش بینی های موجود در این بخش را ذخیره می کند. با داشتن پیش بینی های مدل ها می توانیم خطا ها را محاسبه کنیم:

```
> (nmse.a1.lm <- mean((lm.predictions.a1-algae[, 'a1'])^2)/  
+ mean((mean(algae[, 'a1'])-algae[, 'a1'])^2))
```

```
[1] 0.6473034
```

```
> (nmse.a1.rt <- mean((rt.predictions.a1-algae[, 'a1'])^2)/  
+ mean((mean(algae[, 'a1'])-algae[, 'a1'])^2))
```

```
[1] 0.5942601
```

این آمار آخر این بدی را دارد که نمی تواند در واحد های یکسان قرار بگیرد. آمار جایگزین که پاسخی منطقی فراهم می کند این

آمار محاسباتی را بین اجرای مدل های ما انجام می دهد:

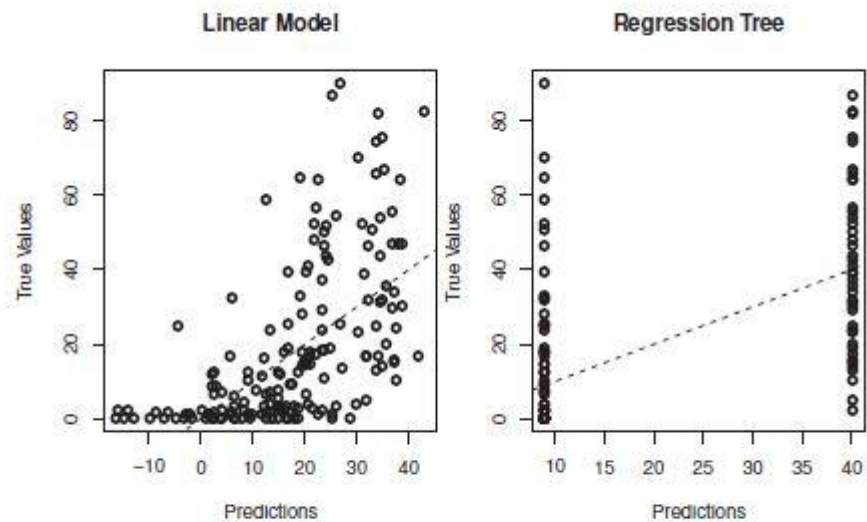


FIGURE 2.10: Errors scatter plot.

NMSE خطایی است که با مقادیر 0 تا 1 اندازه گیری می شود. هرچه این خطا کم تر باشد بهتر است. مقادیر بیش از 1 نشان می

دهد که مدل شما خیلی بد عمل می کند. در زیر می توانید مثالی از استفاده این ساختار پیدا کنید.

```
> old.par <- par(mfrow = c(1, 2))
> plot(lm.predictions.a1, algae[, "a1"], main = "Linear Model",
+      xlab = "Predictions", ylab = "True Values")
> abline(0, 1, lty = 2)
> plot(rt.predictions.a1, algae[, "a1"], main = "Regression Tree",
+      xlab = "Predictions", ylab = "True Values")
> abline(0, 1, lty = 2)
> par(old.par)
```

جالب است که بازرسی تصویری از این پیش بینی ها داشته باشیم. تصویر 2.10 مثالی از این نمونه را نشان می دهد و به صورت کد

زیر ایجاد می شود:

```
> plot(lm.predictions.a1,algae[, 'a1'],main="Linear Model",  
+       xlab="Predictions",ylab="True Values")  
> abline(0,1,lty=2)  
> algae[identify(lm.predictions.a1,algae[, 'a1']),]
```

با نگاه کردن به تصویر 2.10 می توانی م مدل ها را مشاهده کنیم. باید این را به خاطر داشته باشیم که این مدل جدید آسان تر

است. این فرایند تا زمانی که هیچ عامل دیگری وجود نداشته باشد ادامه دارد. کد مقابل مدل خطی که از به کاربردن روش حذف

عقب گردان ایجاد می شود:

```

> sensible.lm.predictions.a1 <- ifelse(lm.predictions.a1 <
+   0, 0, lm.predictions.a1)
> regr.eval(algae[, "a1"], lm.predictions.a1, stats = c("mae",
+   "mse"))

      mae      mse
13.10681 295.54069

> regr.eval(algae[, "a1"], sensible.lm.predictions.a1, stats = c("mae",
+   "mse"))

      mae      mse
12.48276 286.28541

```

در یک سناریو ایده‌آل همه دایره‌ها در جایی که باید باشند قرار می‌گیرند که با ساختاری خاص به دست می‌آیند. هر دایره‌ای در

جایگاه خود هماهنگی‌اش را به دست می‌آورد و مقدار اصلی متغیر را هم چنین. اگر این مقادیر برابر باشد همه این دایره‌ها با در

خطی ایده‌آل قرار بگیرند. ما می‌توانیم نمونه‌ای را که به خوبی پیش‌بینی نشده را حدس بزنیم. که به ما اجازه می‌دهد که روی نقاط

موجود در گراف کلیک کنیم:


```

> cv.rpart <- function(form,train,test,...) {
+   m <- rpartXse(form,train,...)
+   p <- predict(m,test)
+   mse <- mean((p- resp(form,test))^2)
+   c(nmse=mse/mean((mean(resp(form,train))-resp(form,test))^2))
+ }
> cv.lm <- function(form,train,test,...) {
+   m <- lm(form,train,...)
+   p <- predict(m,test)
+   p <- ifelse(p < 0,0,p)
+   mse <- mean((p- resp(form,test))^2)
+   c(nmse=mse/mean((mean(resp(form,train))-resp(form,test))^2))
+ }

```

با استفاده از این کد و پس از اتمام این عملیات با شکل موجود و با راست کلیک روی گراف باید ستون های داده ها را ببینیم زیرا

ما در حال استفاده از بردار هستیم. با توجه به تصویر 2.10 با پیش بینی مدل خطی می توانیم این مدل را پیش بینی کنیم. به همین شکل

می توانیم از این دانش استفاده کنیم و از کمترین آن برای توسعه اجرای مدل خطی استفاده کنیم:

IranDataMiner.ir

CROSS VALIDATION EXPERIMENTAL COMPARISON

** DATASET :: a1

++ LEARNER :: cv.lm variant -> cv.lm.defaults

Repetition 1

Fold: 1 2 3 4 5 6 7 8 9 10

Repetition 2

Fold: 1 2 3 4 5 6 7 8 9 10

Repetition 3

Fold: 1 2 3 4 5 6 7 8 9 10

++ LEARNER :: cv.rpart variant -> cv.rpart.v1

Repetition 1

Fold: 1 2 3 4 5 6 7 8 9 10

Repetition 2

Fold: 1 2 3 4 5 6 7 8 9 10

Repetition 3

Fold: 1 2 3 4 5 6 7 8 9 10

++ LEARNER :: cv.rpart variant -> cv.rpart.v2

Repetition 1

Fold: 1 2 3 4 5 6 7 8 9 10

Repetition 2

Fold: 1 2 3 4 5 6 7 8 9 10

Repetition 3

Fold: 1 2 3 4 5 6 7 8 9 10

++ LEARNER :: cv.rpart variant -> cv.rpart.v3

Repetition 1

Fold: 1 2 3 4 5 6 7 8 9 10

Repetition 2

Fold: 1 2 3 4 5 6 7 8 9 10

Repetition 3

Fold: 1 2 3 4 5 6 7 8 9 10

از این ساختار برای رسیدن به این تأثیرات استفاده می کنیم. این ساختار سه اختلاف دارد. براساس اجرای اندازه گیری های محاسبه

شده یکی باید درخت را برای رسیدن به پیش بینی های 140 نمونه ترجیح دهد. هدف ما انتخاب بهترین مدل برای رسیدن به پیش بینی

هاست کلید این مسئله رسیدن به حدسی مطمئن است برای اجرای مدل ها. مدل هایی وجود دارد که به راحت یبه پیش بینی صفر می

رسد برای انتخاب مدل باید به فرضیاتی قابل اعتماد در مورد داده های دیده نشده برسیم. این روش به طور خلاصه به شکل زیر

توصیف می شود برا یهر یک از داده ها مدلی را با استفاده از ساختاری می سازیم و این مدل را ارزیابی می کنیم. در نهایت ما اندازه

مدله را داریم و با استفاده از آن ها به مدلی در داده ها می رسیم که برای این ساختار نیست و این کلید مسئله است. گاهی اوقات این

ساختار ها را بار ها برای حدس های مطمئن تر تکرار می کنیم. به طور کلی می توان گفت که در برخورد با یک پیش بینی باید

تصمیمات زیر را به کار بندیم:

- مدلی را به عنوان جایگزین برای نشان دادن پیش بینی ها استفاده کنیم
- از روش ارزیابی متری برای مقایسه مدل ها استفاده می کنیم
- استفاده از روش شناسی برای رسیدن به حدسیات مطمئن

این ساختار دارای سه پارامتر است: (1) داده های قابل استفاده در مقایسه (2) مدل های جایگزین (3) پارامتر های آزمایشی .

این ها در مدل خطی به تصویر خواهیم کشید. هر یک از این ساختار ها باید ارزیابی را کامل کند و این ساختار ها باید آزمایشات

را بگذرانند. ساختار ها باید برداری را با مفادیر ارزیابی برگرداند. در این جا دو تا از این ساختار ها را ساخته ایم:

```

> summary(res)

== Summary of a Cross Validation Experiment ==

 3 x 10 - Fold Cross Validation run with seed = 1234

* Datasets :: a1
* Learners :: cv.lm.defaults, cv.rpart.v1, cv.rpart.v2, cv.rpart.v3

* Summary of Experiment Results:

-> Datatataset: a1

      *Learner: cv.lm.defaults
            nmse
avg      0.7196105
std      0.1833064
min      0.4678248
max      1.2218455
invalid 0.0000000

      *Learner: cv.rpart.v1
            nmse
avg      0.6440843
std      0.2521952
min      0.2146359
max      1.1712674
invalid 0.0000000

      *Learner: cv.rpart.v2
            nmse
avg      0.6873747
std      0.2669942

```

در این مثال تصویری به این اشاره می کنیم که از این ارزیابی استفاده می کنیم. پارامترهای باقی مانده شامل پارامترهای ارزیابی شده

می شود. هر دو پارامترها به نتیجه حاصل از بردار ختم می شود. که به ساختارها اجازه می دهد که با متغیرهای خود نمایانگر

شوند. این ساختار تحت تاثیر مجموعه ای از اختلافات است که بعد از اولین درخت از ساختار عبور می کند. خصوصیت دیگر این

ساختارها استفاده از ساختار دیگر یاست که در پکیج ما موجود است و برای رسیدن به مقادیر متغیرها از فرمول داده شده استفاده

می کنیم. کما توجه تعریف ساختارها یادگیری و تست این مدل ها را انجام می دهیم. می توان این مقایسه را به شکل زیر انجام داد:

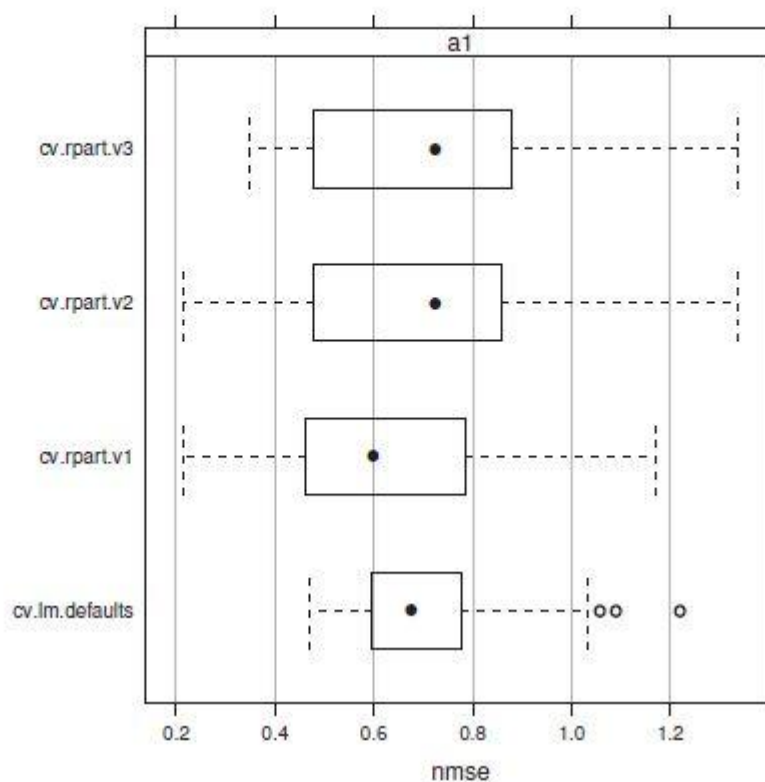


FIGURE 2.11: Visualization of the cross-validation results.

```
min    0.2146359
max    1.3356744
invalid 0.0000000
```

```
*Learner: cv.rpart.v3
```

```
nmse
```

```
avg    0.7167122
std    0.2579089
min    0.3476446
max    1.3356744
invalid 0.0000000
```


IranDataMiner.ir

همان طور که قبلا گفته شد اولین اختلاف باید برداری با داده های استفاده شده در مقایسه های آزمایشی باشد. دومین اختلاف شامل

بردار یادگیر متغیر ها است. هر یک از اختلافات گزیده ای مجموعه ای از ارزش های جایگزین را با مقادیر پارامتری تشکیل

می دهد. در این مثال از روشی با پارامتر هایش استفاده می کنیم. این به این معنا است که آزمایشات شامل سه متغیر است. سومین

پارامتر تنظیمات آزمایشی را تعیین می کند. این پارامتر آخر برای مطمئن ساختن احتمال انجام آزمایشات است. نتیجه این ها شکلی

پیچیده است که شامل همه اطلاعات درگیر با مقایسه ها است. برای مثال کد مقابل خلاصه ای از نتایج مقایسه را ایجاد می کند:

```
> DSs <- sapply(names(clean.algae)[12:18],
+               function(x,names.attrs) {
+                 f <- as.formula(paste(x,"~ ."))
+                 dataset(f,clean.algae[,c(names.attrs,x)],x)
+               },
+               names(clean.algae)[1:11])
> res.all <- experimentalComparison(
+               DSs,
+               c(variants('cv.lm'),
+                 variants('cv.rpart',se=c(0,0.5,1))
+               ),
+               cvSettings(5,10,1234))
```

همان طور که دیده می شود یکی از متغیر ها به بهترین نمره میانگین رسیده است. ما همچنین به این نتایج تصویر سازی رسیده ایم:

```
> plot(res.all)
```

این ساختار برجستگی را بر هر مدلی از متغیرها می چسبانند. اگر بخواهید بدانید که تنظیمات پارامتر به چه شکل است این را دنبال

کنید:

IranDataMiner.ir

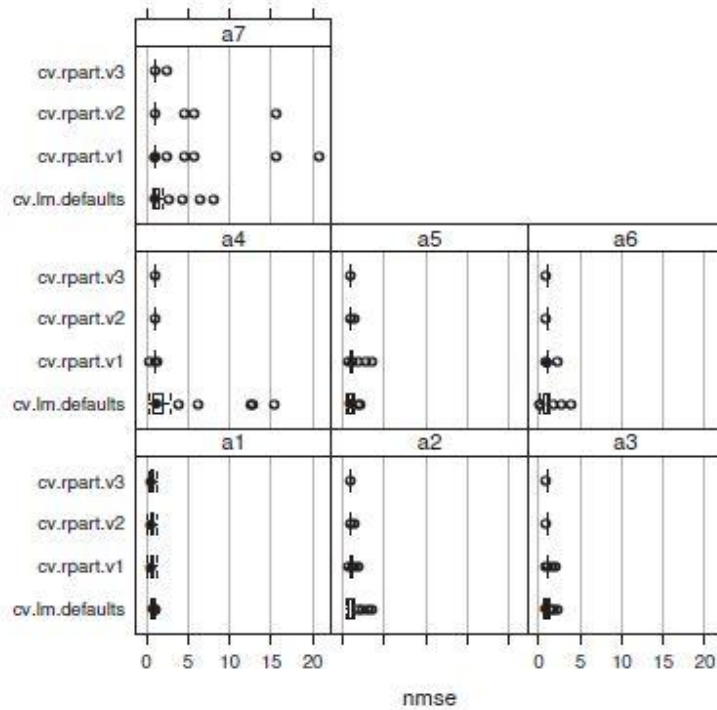


FIGURE 2.12: Visualization of the cross-validation results on all algae.

```
$a2
      system score
nmse cv.rpart.v3    1

$a3
      system score
nmse cv.rpart.v2    1

$a4
      system score
nmse cv.rpart.v2    1

$a5
      system      score
nmse cv.lm.defaults 0.9316803

$a6
      system      score
nmse cv.lm.defaults 0.9359697

$a7
```

برای دلایل خاص برون داد بالا را حذف کرده ایم. برای این منظور باید فرمولی برای هر مشکل بسازیم. این به شکل فرمول دیگری

تغییر یافته است بنابر قدرت کامپیوتر این کد عمل میکند. در تصویر 2.12 نتایج این مدل را برای داده های متفاوت نشان می دهیم.

همان طور که می بینیم دلایل بسیاری وجود دارد که نشان می دهد که میانگین همواره برای متممی متغیر ها قابل اعمال است. برای آن

که ببینیم که کدام مدل بهتر است از این ساختار استفاده می کنیم:

```

> library(randomForest)
> cv.rf <- function(form,train,test,...) {
+   m <- randomForest(form,train,...)
+   p <- predict(m,test)
+   mse <- mean((p-resp(form,test))^2)
+   c(nmse=mse/mean((mean(resp(form,train))-resp(form,test))^2))
+ }
> res.all <- experimentalComparison(
+   DSs,
+   c(variants('cv.lm'),
+     variants('cv.rpart',se=c(0,0.5,1)),
+     variants('cv.rf',ntree=c(200,500,700))
+   ),
+   cvSettings(5,10,1234))

```

این برون داد نشان می دهد که نتایج داده ها امید بخش نیست. مناظر کلی روش های مدلی هستند که سعی به کنار آمدن با محدودیت

هایی را دارد. این ها نظریاتی است برای رسیدن به مناظر کلی متفاوت با مدل ها. جنگل های تصادفی بهترین مثال برای مناظر کلی

است. این ها با مجموعه ای از درخت ها شکل گرفته اند. پیش بینی ها در این باره برای دست یافتن به میانگین درخت ها است. این

پکیج چنین ایده هایی را تکامل می بخشد. کد مقابل آزمایش گذشته را تکرار می کند. دوباره برون داد را برای دلایل خاصی حذف می


```
STD 0.1736676 0.1639373      0.2399321      0.2397636
      Learn.5 sig.5  Learn.6 sig.6  Learn.7 sig.7
AVG 0.6875212  ++ 0.5490511      0.5454724
STD 0.2348946      0.1746944      0.1766636
```

- Dataset: a2

```
      Learn.1  Learn.2 sig.2  Learn.3 sig.3  Learn.4 sig.4
AVG 0.7777851 1.0449317  ++ 1.0426327  ++ 1.01626123  ++
STD 0.1443868 0.6276144      0.2005522      0.07435826
      Learn.5 sig.5  Learn.6 sig.6  Learn.7 sig.7
AVG 1.000000e+00  ++ 0.7829394      0.7797307
STD 2.389599e-16      0.1433550      0.1476815
```

- Dataset: a4

```
      Learn.1  Learn.2 sig.2  Learn.3 sig.3  Learn.4 sig.4
AVG 0.9591182 2.111976      1.0073953  + 1.000000e+00  +
STD 0.3566023 3.118196      0.1065607      2.774424e-16
      Learn.5 sig.5  Learn.6 sig.6  Learn.7 sig.7
AVG 1.000000e+00  + 0.9833399      0.9765730
STD 2.774424e-16      0.3824403      0.3804456
```

- Dataset: a6

```
      Learn.1  Learn.2 sig.2  Learn.3 sig.3  Learn.4 sig.4
AVG 0.9126477 0.9359697  ++ 1.0191041      1.000000e+00
STD 0.3466902 0.6045963      0.1991436      2.451947e-16
      Learn.5 sig.5  Learn.6 sig.6  Learn.7 sig.7
AVG 1.000000e+00      0.9253011      0.9200022
STD 2.451947e-16      0.3615926      0.3509093
```

Legends:

Learners -> Learn.1 = cv.rf.v3 ; Learn.2 = cv.lm.defaults ;
 Learn.3 = cv.rpart.v1 ; Learn.4 = cv.rpart.v2 ; Learn.5 = cv.rpart.v3 ;
 Learn.6 = cv.rf.v1 ; Learn.7 = cv.rf.v2 ;
 Signif. Codes -> 0 '++' or '---' 0.001 '+' or '-' 0.05 ' ' 1

برای متممی مشکلات به جز داده 7 بهترین امتیاز به دست آمده مربوط به جنگل تصادفی است. برون داد این ساختار تفاوت بین

امتیاز های موجود در مدل ها را به ما نمی گوید. این ساختار در پکیج ما این اطلاعات را فراهم می کند و مجموعه ای از تست ها

را بین مدل ها و سایر جایگزین ها ایجاد می کند. این مدل بهترین است برای داده های 1 و 2 و 4 و 6. کنترل روبرو به این شکل است:

```

> compAnalysis(res.all,against='cv.rf.v3',
               datasets=c('a1','a2','a4','a6'))

== Statistical Significance Analysis of Comparison Results ==

Baseline Learner::          cv.rf.v3  (Learn.1)

** Evaluation Metric::      nmse

- Dataset: a1
      Learn.1  Learn.2 sig.2  Learn.3 sig.3  Learn.4 sig.4
AVG 0.5447361 0.7077282  ++ 0.6423100  + 0.6569726  ++

```

ستون 'isg.X' اطلاعاتی را که ما به دنبالش هستیم فراهم می کند. نبود نشانه به این معناست که اطمینانی در مشاهده تفاوت ها وجود

ندارد. سیگنال ها به این معنا هستند که ارزیابی میانگینی مدل ها از این ساختار بالاتر است. تفاوت بی این متغیر های جنگل تصادفی و

سایرین به صورت آماری دقیق نیست. با توجه به سایر مدل ها در بیشتر موارد منفعتی مهم برای این متغیر های تصادفی وجود دارد. ما

تحلیلی مشابه را برای سایر مدل های که بهترین امتیاز را دارند با استفاده از مقادیر مختلف انجام داده ایم. پیش بینی های هفت خزه

در این بخش می بینیم که چگونه به پیش بینی های هفت خزه در 140 نمونه می رسیم. فرایند طی شده شامل فرضیات بی پایه برای

گروهی از مدل ها به وسیله پردازش آزمایشی می شود. هدف اصلی در این داده ها رسیدن به این هفت خزه در 140 نمونه است. این

یکی از مدل های نشان داده شده در ساختار قبلی است. بیایید با رسیدن به این مدل ها با استفاده از مدل های موجود از تست آن ها

استفاده کنیم. این موضوع می تواند از برگشت درختی به عنوان روشی برای مقادیر ناشناخته جلوگیری کند. جنگل های تصادفی به

طور موقت شامل چنین روشی نمی شود. کد مقابل به همه هفت مدل می رسد:

```
> bestModelsNames <- sapply(bestScores(res.all),  
+                             function(x) x['nmse','system'])  
> learners <- c(rf='randomForest',rpart='rpartXse')  
> funcs <- learners[sapply(strsplit(bestModelsNames,'\\.'),  
+                             function(x) x[2])]  
> parSetts <- lapply(bestModelsNames,  
+                     function(x) getVariant(x,res.all)@pars)  
> bestModels <- list()  
> for(a in 1:7) {
```

ما با استفاده از یک بردار با نام های متغیرها شروع کردیم. و این با گسترش قسمتی از نام متغیری به دست آمده است. این لیست با

تنظیمات پارامتری برای هر یک از متغیرها امضا شده است. شکل با این ساختار برگشته است. شکاف های اشکال به واسطه اپراتور

ایجاد می شود. در نهایت به این مدل ها می رسیم و و آن ها را ذخیره می کنیم. و با استفاده از ساختاری که به ما اجازه می دهد از هر

ساختاری استفاده کنیم ایجاد می شود. صفت اطلاعات موجود در پکیج شامل 140 نمونه است. اولین محرک برای ایجاد این عمل اجرای

ساختار صفحه کلید است. با به کار بردن این ساختار به طور مستقیم این مجموعه را آزمایش می کنیم. اگرچه از داده ها استفاده نمی

کنیم اما آن چه که اشتباه است آن است که از این فرآیند جلوگیری می کنیم. این ساختار اختلافی دیگر دارد که می تواند در این موقعیت

ها استفاده شود. از این استفاده می کنیم:

```
+ form <- as.formula(paste(names(clean.algae)[11+a], '~ .'))
+ bestModels[[a]] <- do.call(funcs[a],
+   c(list(form, clean.algae[, c(1:11, 11+a)]), parSetts[[a]]))
+ }
```

این اختلاف داده ای این امکان را می دهد که داده های دیگری را در کنار آن ها با مقادیر ناشناخته ایجاد کنیم. توجه کنید که ما این را

از صفحه داده ها حذف کرده ایم. اکنون آماده ایم که به ماتریس داده هادست یابیم:

```
> preds <- matrix(ncol=7, nrow=140)
> for(i in 1:nrow(clean.test.algae))
+   preds[i,] <- sapply(1:7,
+     function(x)
+       predict(bestModels[[x]], clean.test.algae[i,])
+   )
```

با این کد ساده به ماتریس 140×7 می‌رسیم. در این مکان پیش‌بینی‌ها را با هم مقایسه می‌کنیم. مقدار اصلی آزمایش در این صفحه

داده وجود دارد. کد مقابل امتیازات را برای مدل‌ها محاسبه می‌کند:

```
> avg.preds <- apply(algae[,12:18],2,mean)
> apply( ((algae.sols-preds)^2),          2,mean) /
+ apply( (scale(algae.sols,avg.preds,F)^2),2,mean)

      a1      a2      a3      a4      a5      a6      a7
0.4650380 0.8743948 0.7798143 0.7329075 0.7308526 0.8281238 1.0000000
```

ابتدائاً به پیش‌بینی‌ها در مدل برای محاسب‌ه‌که شامل میانگین متغیرها می‌رسیم. این امر با عبارتی ساده که شاید قدری پیچیده به

نظر برسد اما خیلی زود حل می‌شود انجام می‌شود. یان ساختار برای نرمال کردن داده‌ها مورد استفاده قرار می‌گیرد. در این مثال

از بردار گسترده استفاده کرده ایم. نتایج که به آن‌ها رسیده ایم در برخورد با متغیرها فرضیه است. به طور خلاصه با مدلی مناسب

می‌توانیم به امتیازات جالبی برای این مشکلات برسیم.

خلاصه

هدف اولیه این مطالعات آشنا کردن خواننده با R است. برای این هدف از مشکلی کوچک حداقل با استانداردهای داده‌ها استفاده

کرده ایم. چگونگی اجرای برخی از داده های اساسی را توصیف کرده ایم. اگر خواهان دانستن بیشتر درباره این داده ها هستید می

توانید از سایت ها یا برخی روزنامه ها استفاده کنید. در باره داده های استخراجی این مطالعات اطلاعاتی تهیه شده است درباره

- تصویرسازی داده ها
- آمار های توصیفی
- راه کار های کنترل مقادیر ناشناخته متغیر ها
- عملیات برگشت
- ارزیابی متری برای عملیات برگشتی
- برگشت خطی چند بعدی
- درخت های برگشتی
- مدل انتخابی/مقایسه از راه k-fold در برابر ارزش
- مناظر کلی مدل و جنگل های تصادفی

امیدواریم که از این به بعد با عملکرد R عجین شوید و با برخی از این چهره ها آشنا شوید. برای مثال بای د این تکنیک ها را یاد گرفته

باشید

- ذخیره اطلاعات از روی فایل های متنی
- چگونگی دست یافتن به آمار های توصیفی
- تصویرسازی اساسی داده ا
- کنترل داده ها با مقادیر ناشناخته
- چگونگی دسترسی به مدل های برگشتی
- چگونگی استفاده از مدل ها برای رسیدن به جزئیات این داده ها

مطالعات بیشتر به شما جزئیاتی راجع به این تکنیک ها می دهد.

فصل سوم

مترجم: احسان رستمی

پیش بینی بازده بازار سهام

این مطالعه (دومین مطالعه) سعی دارد کمی بیشتر در مورد تعریف واژه استفاده از روش داده کاوی پیش رود. ما به برخی از مشکلات یکی کردن ابزارها و فناوری‌های داده کاوی در بهم پیوستن مشکلات تجاری اشاره می‌کنیم. فضاهاى اینترنتى خاصى براى توضیح دادن این مشکلات استفاده می‌شود که یکی از آن سیستم‌های مبادله خودکار سهام است. ما به وظیفه ساختن یک سیستم مبادله سهام بر مبنای مدل پیشگویی برگرفته از اطلاعات قیمت جاری سهام روزانه خواهیم پرداخت. چندین مدل با هدف پیش‌بینی بازده آتی 500 شاخص مورد بررسی قرار خواهد گرفت. این پیش‌بینی‌ها با یک استراتژی تجاری توأمان جهت دستیابی به S&P بازار یک تصمیم مربوط به سفارشات بازار برای تولید استفاده خواهد شد. این فصل چندین نظریه داده کاوی جدید از برای تجزیه و تحلیل داده‌های ذخیره شده در دیتابیس استفاده کنیم (2) چگونه R جمله اینکه (1) چگونه از مشکلات پیش‌بینی را بوسیله زمان سفارش در بین داده‌های مشاهده شده (که به عنوان سری زمانی شناخته می‌شود) مدیریت کنیم و (3) یک مثال از مشکلات پیش‌بینی‌های مدل ترجمه به تصمیم و عملکرد را در برنامه‌های دنیای واقعی، را مورد بررسی قرار می‌دهد.

3-1- تشریح مسئله و اهداف آن

معاملات بازار سهام یک قلمرو برنامه‌نویسی با پتانسیل بالایی جهت داده کاوی است. در نتیجه، وجود مقادیر عظیمی از داده‌های تاریخی پیشنهاد می‌کند که داده کاوی می‌تواند یک مزیت رقابتی فراتر از انتظارات انسانی در

مورد این داده‌ها فراهم سازد. از سوی دیگر، محققان ادعا دارند که بازارها به سرعت با شرایط تنظیم قیمت وفق می‌یابند بطوریکه هیچ فضایی برای دستیابی به سود به شیوه‌ای ثابت وجود ندارد. این مسئله عموماً تحت عنوان فرضیه بازارهای کارا (بهره‌ور) شناخته می‌شود. این تئوری به طور موفقیت‌آمیزی به وسیله نسخه‌های آسان‌تر که برای شانس‌های مبادلاتی مقرر جهت شرایط ناکارآمدی موقت بازار فضاهایی باقی می‌گذارد، جایگزین شد.

هدف کلی از مبادله سهام نگهداری و حفظ یک سند از مبنای دارایی‌ها در سفارشات خرید و فروش است. هدف بلند مدت دستیابی به بیشترین سود ممکن از انجام این مبادلات تجاری است. در متن این فصل این سناریوی کلی را کمی بیشتر مورد بحث قرار می‌دهیم. برای مثال، "تجارت" یک امنیت، یا در حقیقت یک شاخص بازار است. با توجه به این امنیت و یک سرمایه اولیه، ما سعی خواهیم کرد سودمان را در یک دوره آزمایشی آتی بوسیله مفهوم مبادلات تجاری (خرید، فروش، نگهداری) حداکثر کنیم. استراتژی تجاری ما به عنوان یک مبنا برای تصمیم‌گیری در مورد آثار بوسیله نتیجه حاصل از فرایند داده‌کاوی فراهم می‌کند. این فرایند شامل سعی در پیش‌بینی ارزش آتی شاخص بر مبنای یک مدل بدست آمده با اطلاعات تاریخی خواهد بود. بنابراین مدل پیش‌بینی ما در یک سیستم تجاری که مبنای تصمیماتش را بر پیش‌بینی‌هایی که از مدل بوجود می‌آید، ثبت خواهد شد. ضوابط ارزیابی کلی ما عملکرد این سیستم تجاری خواهد بود، که نتیجه سود/زیان از عملکرد سیستم به خوبی بعضی آمار دیگر است و مورد علاقه سرمایه‌گذاران می‌باشد. این بدین معناست که ضابطه ارزیابی اصلی ما نتایج عملیاتی از کاربرد اطلاعات کشف شده بوسیله فرایند داده‌کاوی مان خواهد بود و نه دقت و صحت مدل‌های توسعه یافته در خلال این فرایند.

3_2_ اطلاعات در دسترس

تمرکز خواهیم کرد. اطلاعات روزانه در ارتباط با نقل این S&P در مطالعه‌مان بر مبادله شاخص بازار 500 امنیت به طوری آزادانه در بسیاری مکان‌ها در دسترس است، برای مثال سایت مالی یاهو.

اطلاعاتی که ما استفاده خواهیم کرد در بسته کتاب در دسترس است. یک بار دیگر ما مفهوم‌های دیگری از را جستجو خواهیم کرد. بعلاوه، برخی R بدست آوردن اطلاعاتی مانند یک فرم از توضیحات برخی ظرفیت‌های از دیگر از پیشنهادات متناوب به شما برای کاربرد مفاهیم آموخته شده در این فصل برای اطلاعات جاری بیش از یک پکیج در زمان نوشته شدن این کتاب اجازه خواهد داد.

کتاب، این کافیه که اشاره کنیم: R به عبارت دیگر برای بدست آوردن اطلاعات از طریق پکیج

```
> library(DMwR)
```

```
> data(GSPC)
```

اشاره نکرده باشید. دستورالعمل R اولین عبارت فقط وقتی مورد نیاز است که شما در مورد آن قبل از بخش بازخوانی خواهد کرد. ما این اهداف این کلاس را در بخش 1-2-3 مورد xts، را از کلاس GSPC دوم یک هدف، بحث قرار خواهیم داد اما فعلا شما می‌توانید آن را دستکاری کنید چنانکه این ماتریس یا فریم اطلاعات است (GSPC) (سعی کنید، برای مثال، هد

است که با (CSV) در وب سایت کتاب، می‌توانید این اطلاعات را در دو فرمت دیگر بیابید. اولی یک فایل به شیوه‌ای مشابه که اطلاعات در فصل 2 استفاده شده است. R کاما جدا شده است که می‌تواند خوانده شود در است که ما می‌توانیم برای خلق کردن یک دیتابیس با نقل قول MySQL فرمت دیگر یک فایل موقت دیتابیس برای این دو فرمت R استفاده کنیم. ما توضیح خواهیم داد چگونه این اطلاعات بوسیله MySQL 500 در S&P دیگر لود خواهد شد. این به شما بستگی دارد که تصمیم بگیرید با چه تناوبی شما دانلود خواهید کرد، یا اگر شما ترجیح می‌دهید به شیوه آسانی آنرا از پکیج کتاب بارگذاری کرده و استفاده کنید. باقیمانده فصل (برای مثال تجزیه و تحلیل بعد از خواندن داده‌ها) از الگوی ذخیره شده‌ای که شما تصمیم دارید استفاده کنید مستقل است.

اشاره خواهیم کرد، که R برای تکامل ما همچنین هنوز به راه دیگری از بدست آوردن این اطلاعات در شامل دانلود مستقیم آن از وب سایت است. اگر انتخاب کردید که این راه را دنبال کنید، باید به خاطر داشته باشید که شما احتمالا یک مجموعه داده بزرگتری از آنکه در تجزیه و تحلیل انجام شده در این کتاب آمده، استفاده خواهید کرد.

هر منبعی که شما برای استفاده انتخاب کنید اطلاعات منقول از سهام روزانه شامل اطلاعات جمع‌آوری شده در موارد ذیل است:

- تاریخ جلسه مبادله سهام
- قیمت آغازین در ابتدای جلسه
- بیشترین قیمت در خلال جلسه
- کمترین قیمت
- قیمت بسته شدن جلسه
- حجم مبادلات

- قیمت بسته تعدیل شده

R₁₋₂₋₃ مدیریت اطلاعات وابسته به زمان در

اطلاعات در دسترس برای این مطالعه به زمان بستگی دارد. بدین معنی که هر مشاهده‌ای از اطلاعات یک برچسب زمانی دارد که به آن متصل است. این مدل از اطلاعات به طور مکرر به عنوان سری‌های زمانی شناخته می‌شوند. مشخصه اصلی این نوع از داده‌ها این است که بین موضوعات، طبق برچسب زمانی الصاق شده به آن‌ها است: y سفارش داده می‌شود. به طور کلی، یک سری زمانی یک سری مشاهدات سفارش داده شده از یک متغیر

$$y_1, y_2, \dots, y_{t-1}, y_t, y_{t+1}, \dots, y_n \quad (1-3)$$

است می‌باشد. t در زمان y ارزش متغیر y_t جایگاه

، y_1, y_2 هدف اصلی تجزیه و تحلیل سری‌های زمانی یافتن یک مدل بر مبنای مشاهدات گذشته متغیرهای پیش‌بینی کنیم. y_n, \dots, y_{t+1} است که به ما اجازه می‌دهد راجع به مشاهدات آتی متغیر y_t, y_{t-1}, \dots

در رابطه با اطلاعات سهام، چیزی داریم که معمولاً به عنوان سری‌های زمانی چند متغیره شناخته می‌شود، *Open, High, Low* زیرا ما چندین متغیر را در برچسب زمانی واحدی اندازه‌گیری می‌کنیم، برای مثال *Close, Volume* و *AdjClose*.

چندین پکیج اختصاص یافته به تجزیه و تحلیل این مدل از داده دارد و در نتیجه آن، کلاس‌های خاصی R توابع تنظیم شده بسیاری برای R از اهدافی که برای اطلاعات وابسته ذخیره شده استفاده می‌شود دارد. بعلاوه، این مدل از اهداف، مانند توابع نقشه‌کشی ویژه و غیره دارد.

در بین پکیج‌های *xts* (Ryan and Ulrich, 2010) و *zoo* (Zeileis and Grothendieck, 2005) برای مدیریت اطلاعات وابسته به زمان هستند. هر دو قدرت مشابهی را پیشنهاد می‌دهند، R انعطاف‌پذیرتر یک سری از امکانات اضافی (برای مثال، اصطلاحی از تنظیمات فرعی استفاده شده در رشته زمانی *xts* اگرچه را *zoo* کلاس *xts*) برای مدیریت کردن این مدل از اطلاعات را فراهم می‌کند. از لحاظ فنی کلاس *ISO 8601* است و بنابراین ما می‌توانیم هر روش *zoo* همچنین یک هدف *xts* گسترش می‌دهد، بدین معنی که هر هدف نیز استفاده کنیم. ما تجزیه تحلیل در این فصل را اصولاً *xts* بکار می‌بریم برای اهداف *zoo* طراحی برای اهداف پایه‌گذاری خواهیم کرد. با مثال‌های گویای اندکی از خلق و استفاده این مدل از هدف آغاز *xts* بر اهداف

پکیج‌های اضافی هستند (برای مثال اینکه با نصب `xts` و `zoo` می‌کنیم. لطفا توجه داشته باشید که هر دوی نیاز دارید (به بخش 1_2_1 صفحه 3 را مراجعه R نمی‌آید) و اینکه شما به دانلود کردن و نصب آن در R پایه کنید).

ایجاد می‌شود. `xts` مثال‌های زیر شرح می‌دهند که چگونه اهداف کلاس

```
> library(xts)
> x1 <- xts(rnorm(100), seq(as.POSIXct("2000-01-01"), len = 100,
+   by = "day"))
> x1[1:5]
```

```
      [,1]
2000-01-01 0.82029230
2000-01-02 0.99165376
2000-01-03 0.05829894
2000-01-04 -0.01566194
2000-01-05 2.02990349
```

```
> x2 <- xts(rnorm(100), seq(as.POSIXct("2000-01-01 13:00"),
+   len = 100, by = "min"))
> x2[1:4]
```

```
      [,1]
2000-01-01 13:00:00 1.5638390
2000-01-01 13:01:00 0.7876171
2000-01-01 13:02:00 1.0860185
2000-01-01 13:03:00 1.2332406
```

```
> x3 <- xts(rnorm(3), as.Date(c("2005-01-01", "2005-01-10",
+   "2005-01-12"))))
> x3
```

```
      [,1]
2005-01-01 -0.6733936
2005-01-10 -0.7392344
2005-01-12 -1.2165554
```

اطلاعات سری‌های زمانی را در اولین متغیر مستقل دریافت می‌کند. این مطلب همچنین (`xts` تابع در صورتیکه ما یک سری زمانی چند متغیره داشته باشیم می‌تواند یک بردار یا یک ماتریس باشد.

در مورد اخیر هر ستون از ماتریس به عنوان یک متغیر که در برچسب زمانی نمونه‌گیری می‌شود، تفسیر می‌گردد (برای مثال هر ردیف). برچسب زمانی بوسیله متغیر مستقل دوم فراهم می‌شود. این نیاز دارد به . در مثال بالا ما دو تا از R برچسب‌های زمانی تنظیم شده در هر کدام از کلاس‌های زمانی موجود در و کلاس POSIXct/POSIXit استفاده کردیم: کلاس‌های R معمولی‌ترین کلاس‌ها را برای ارائه اطلاعات در زمان. توابع بسیاری وجود دارد که با این اهداف جهت دستکاری تاریخ‌های اطلاعات ارتباط دارد که ممکن است یک نمونه از مثال است. ما این تابع را قبل از seq () بخواهید. تابع R شما برای بررسی با استفاده از تسهیلات تولید رشته اعداد بکار می‌بریم. اینجا ما این را برای تولید رشته‌های بر مبنای زمان بکار می‌بریم همچنان که شما در مثال می‌بینید. همچنانکه در مثال‌های کوچک بالا ممکن است مشاهده کنید، اهداف ممکن است فهرست شود همچنانکه اگر آن‌ها اهداف "معمولی" بدون برچسب زمانی هستند (در این مورد ما یک بردار تنظیمات فرعی استاندارد می‌بینیم). هنوز، ممکن به طور مکرر مبنای اهداف سری‌های زمانی را به عنوان زیرمجموعه شرط‌های وابسته به زمان بخواهیم. این مسئله همچنانکه مثال‌های کوچ زیر سعی دارند آن را بیان قابل دستیابی است: xts کنند، در چندین شیوه با اهداف

```
> x1[as.POSIXct("2000-01-04")]
```

```
      [,1]
2000-01-04 -0.01566194
```

```
> x1["2000-01-05"]
```

```
      [,1]
2000-01-05 2.029903
```

```
> x1["20000105"]
```

```
      [,1]
2000-01-05 2.029903
```

```
> x1["2000-04"]
```

```
      [,1]
2000-04-01 01:00:00 0.2379293
2000-04-02 01:00:00 -0.1005608
2000-04-03 01:00:00 1.2982820
```

2000-04-04	01:00:00	-0.1454789
2000-04-05	01:00:00	1.0436033
2000-04-06	01:00:00	-0.3782062
2000-04-07	01:00:00	-1.4501869
2000-04-08	01:00:00	-1.4123785
2000-04-09	01:00:00	0.7864352

> x1["2000-03-27"]

		[,1]
2000-03-27	01:00:00	0.10430346
2000-03-28	01:00:00	-0.53476341
2000-03-29	01:00:00	0.96020129
2000-03-30	01:00:00	0.01450541
2000-03-31	01:00:00	-0.29507179
2000-04-01	01:00:00	0.23792935
2000-04-02	01:00:00	-0.10056077
2000-04-03	01:00:00	1.29828201
2000-04-04	01:00:00	-0.14547894
2000-04-05	01:00:00	1.04360327
2000-04-06	01:00:00	-0.37820617
2000-04-07	01:00:00	-1.45018695
2000-04-08	01:00:00	-1.41237847
2000-04-09	01:00:00	0.78643516

> x1["2000-02-26/2000-03-03"]

	[,1]
2000-02-26	1.77472194
2000-02-27	-0.49498043
2000-02-28	0.78994304
2000-02-29	0.21743473
2000-03-01	0.54130752
2000-03-02	-0.02972957
2000-03-03	0.49330270

> x1["/20000103"]

```
[,1]
2000-01-01 0.82029230
2000-01-02 0.99165376
2000-01-03 0.05829894
```

دستور اول از یک مقدار واقعی از همان کلاس مانند هدف داده شده در متغیر مستقل دوم در زمان ایجاد معرفی شده `xts` استفاده می‌کند. مثال‌های دیگر یک الگوی فهرست سازی قوی را که بوسیله پکیج `x1` می‌باشد. این الگو `R` است را شرح می‌دهد که یکی از مزیت‌های آن برتر بودن از سایر پکیج‌های سری زمانی در اجرا می‌کند. `[.s] CCYY-MM-DD HH:MM:SS` برچسب‌های زمان را همچون رشته‌هایی با فرمت کلی همچنان که می‌توانید در مثال‌ها تایید کنید، جدا کننده‌ها می‌تواند حذف شود و بخشی از ویژگی‌های زمانی `"/"` برای در بر گرفتن مجموعه‌هایی از برچسب‌های زمان رها شده و نادیده گرفته شود. علاوه بر این، علامت می‌تواند برای فاصله‌های زمانی خاصی که ممکنه در هر دو طرف (هر دو انتها) با مفهومی از شروع یا پایان برچسب زمان مشخص نشده، استفاده شود. سری‌های زمانی چندگانه می‌تواند در یک روش مشابه آنچنان که در زیر شرح داده می‌شود ایجاد گردد:

```
> mts.vals <- matrix(round(rnorm(25),2),5,5)
> colnames(mts.vals) <- paste('ts',1:5,sep='')
> mts <- xts(mts.vals,as.POSIXct(c('2003-01-01','2003-01-04',
+ '2003-01-05','2003-01-06','2003-02-16'))))
> mts
```

	ts1	ts2	ts3	ts4	ts5
2003-01-01	0.96	-0.16	-1.03	0.17	0.62
2003-01-04	0.10	1.64	-0.83	-0.55	0.49
2003-01-05	0.38	0.03	-0.09	-0.64	1.37
2003-01-06	0.73	0.98	-0.66	0.09	-0.89
2003-02-16	2.68	0.10	1.44	1.37	-1.37

```
> mts["2003-01",c("ts2","ts5")]
```

	ts2	ts5
2003-01-01	-0.16	0.62
2003-01-04	1.64	0.49
2003-01-05	0.03	1.37
2003-01-06	0.98	-0.89

و () ارزش‌های اطلاعات سری‌های زمانی بدست می‌آید، از شاخص توابع () در حالیکه از تابع دیتای مرکزی استفاده می‌شود. xts می‌تواند برای "جداسازی" اطلاعات برچسب‌های زمان هر شی () زمان

> index(mts)

[1] "2003-01-01 WET" "2003-01-04 WET" "2003-01-05 WET" "2003-01-06 WET"

[5] "2003-02-16 WET"

> coredata(mts)

	ts1	ts2	ts3	ts4	ts5
[1,]	0.96	-0.16	-1.03	0.17	0.62
[2,]	0.10	1.64	-0.83	-0.55	0.49
[3,]	0.38	0.03	-0.09	-0.64	1.37
[4,]	0.73	0.98	-0.66	0.09	-0.89
[5,]	2.68	0.10	1.44	1.37	-1.37

برای ذخیره اطلاعات قیمت جاری سهام کافی و مناسب است، همچنانکه آن‌ها xts بطور خلاصه، اشیاء برای ذخیره سری‌های زمانی چندگانه با برچسب‌های زمان نامنظم و فراهم کردن الگوهای فهرست بندی قدرتمندی اجازه دارند.

2-2-3 CSV خواندن فرم اطلاعات از فایل

همچنان که در قبل اشاره کردیم، در کتاب وب سایت شما می‌توانید منابع متفاوتی شامل اطلاعات برای فایل استفاده کنید فایلی که خط اولش شبیه این است CSV استفاده در این مطالعه بیابید. اگر تصمیم دارید از را دانلود خواهید کرد:

"Index"	"Open"	"High"	"Low"	"Close"	"Volume"	"AdjClose"
1970-01-02	92.06	93.54	91.79	93	8050000	93
1970-01-05	93	94.25	92.53	93.46	11490000	93.46
1970-01-06	93.46	93.81	92.13	92.82	11460000	92.82
1970-01-07	92.82	93.38	91.93	92.63	10010000	92.63
1970-01-08	92.63	93.47	91.99	92.68	10670000	92.68
1970-01-09	92.68	93.25	91.82	92.4	9380000	92.4
1970-01-12	92.4	92.67	91.2	91.7	8900000	91.7

در همان دایرکتوری که مربوط به "sp500.csv" فرض می‌شود شما فایل را دانلود کرده و با همان نام با اطلاعاتی مانند زیر ایجاد xts بارگذاری کرده و یک شیء R است ذخیره کرده‌اید، می‌توانید آن را در R بخش کنید:

```
> GSPC <- as.xts(read.zoo("sp500.csv", header = T))
```

zoo را می‌خواند و اطلاعات را به شیء فرضی CSV یک فایل zoo از بسته (پکیج) read.zoo تابع شیء نتیجه شده را به یک شیء از as.xts انتقال می‌دهد که اولین ستون آن شامل برچسب زمان است. تابع به زور تبدیل می‌کند. xts کلاس

3-2-3_ بدست آوردن اطلاعات (داده) از وب

استفاده از خدمات رایگان فراهم شده S&P500 یک راه متناوب دیگر در بدست آوردن ارزش جاری را به فایلی با ارزش جاری که شما CSV بوسیله بخش مالی یاهو است که اجازه می‌دهد شما یک فایل شامل تابع R پکیج (Trapletti and Hornik, 2009) tseries می‌خواهید تبدیل کند. استفاده شود. مطلب زیر یک zoo می‌باشد که می‌تواند برای دانلود ارزش جاری در شیء (get.hist.quote) است: S&P500 مثال استفاده از این تابع برای بدست آوردن ارزش جاری

```
> library(tseries)
> GSPC <- as.xts(get.hist.quote("^GSPC", start="1970-01-02",
  quote=c("Open", "High", "Low", "Close", "Volume", "AdjClose")))
```

```
...
...
```

```
> head(GSPC)
```

	Open	High	Low	Close	Volume	AdjClose
1970-01-02	92.06	93.54	91.79	93.00	8050000	93.00
1970-01-05	93.00	94.25	92.53	93.46	11490000	93.46
1970-01-06	93.46	93.81	92.13	92.82	11460000	92.82
1970-01-07	92.82	93.38	91.93	92.63	10010000	92.63
1970-01-08	92.63	93.47	91.99	92.68	10670000	92.68
1970-01-09	92.68	93.25	91.82	92.40	9380000	92.40

را برای `as.xts()` باز می‌گردد، ما دوباره تابع `get.hist.quote()` همچون تابع `zoo` یک شیء از کلاس استفاده می‌کنیم. ما باید خاطر نشان کنیم که اگر شما این فرمان‌ها را انتشار دادید، `xts` تبدیل کردن آن به اطلاعات بیشتری نسبت به آنچه با شیء فراهم شده در پکیج کتاب به دست خواهید آورد. اگر بخواهید مطمئن باشید که شما نتایج مشابهی در فرمان‌های آتی در این فصل بدست آورده‌اید باید در عوض از فرمان زیر استفاده کنید.

```
> GSPC <- as.xts(get.hist.quote("^GSPC",
  start="1970-01-02",end='2009-09-15',
  quote=c("Open", "High", "Low", "Close", "Volume", "AdjClose")))
```

پکیج ما است. روش دیگر بدست `GSPC` آخرین روز قیمت‌های جاری در شیء "2009-09-15" جاییکه آوردن اطلاعات جاری قیمت از وب (اما این تنها راه نیست همچنانکه در ادامه خواهیم دید)، برای استفاده از است. دوباره این جزو قسمت‌های اضافی (quantmod (Ryan, 2009) از پکیج `getSymbols()` تابع پکیج است که شما باید قبل از استفاده اینستال کنید. این قسمت چندین تسهیلات وابسته به تجزیه و تحلیل در پیوستگی `getSymbols()` اطلاعات مالی فراهم می‌کند که ما در خلال این فصل استفاده خواهیم کرد. تابع با سایر توابع از این پکیج یک راه ساده‌تر اما قدرتمند برای بدست آوردن اطلاعات جاری از منابع اطلاعاتی متفاوت فراهم می‌کند. اجازه دهید بعضی مثال‌های استفاده از آن را ببینیم:

```
> library(quantmod)
> getSymbols("^GSPC")
```

در متغیر مستقل اول یک گروه از علائم و اسم‌ها را دریافت می‌کند و ارزش جاری `getSymbols()` تابع این علامت‌ها را از منابع مختلف وب یا حتی دیتابیس‌های محلی جمع‌آوری خواهد کرد، که به آرامی در محیط با همان نام شبیه علامت بازگردانده می‌شود. تابع `xts` کار ایجاد شده و بوسیله مقادیر پیش‌فرض یک شیء پارامترهای بسیاری دارد که بر بعضی از این نظریات کنترل بیشتری را مقدور می‌سازد. همچنانکه می‌توانید بررسی کنید، شیء بازگردانده شده دوره مشابهی مانند اطلاعات آمده با پکیج کتاب را پوشش نمی‌دهد و ستون نام‌های آن اندکی متفاوت است. این امر می‌تواند به سادگی مانند زیر عملی شود:

```
> getSymbols("^GSPC", from = "1970-01-01", to = "2009-09-15")
> colnames(GSPC) <- c("Open", "High", "Low", "Close", "Volume",
+ "AdjClose")
```

، شما در حقیقت چندین علامت با منابع اطلاعاتی quantmod با چارچوب فراهم شده بوسیله پکیج وابسته متفاوتی دارید، که هر کدام با پارامتر مخصوص خود است. تمام این تنظیمات می‌تواند همچنانکه در مثال مشخص گردد. setSymbolLookup() با تابع R ساده زیر می‌بینید، در ابتدای قسمت

```
> setSymbolLookup(IBM=list(name='IBM',src='yahoo'),
+                 USDEUR=list(name='USD/EUR',src='oanda'))
> getSymbols(c('IBM','USDEUR'))
```

```
> head(IBM)
```

	IBM.Open	IBM.High	IBM.Low	IBM.Close	IBM.Volume	IBM.Adjusted
2007-01-03	97.18	98.40	96.26	97.27	9196800	92.01
2007-01-04	97.25	98.79	96.88	98.31	10524500	93.00
2007-01-05	97.60	97.95	96.91	97.42	7221300	92.16
2007-01-08	98.50	99.50	98.35	98.90	10340000	93.56
2007-01-09	99.08	100.33	99.07	100.07	11108200	94.66
2007-01-10	98.50	99.05	97.93	98.89	8744800	93.55

```
> head(USDEUR)
```

	USDEUR
2009-01-01	0.7123
2009-01-02	0.7159
2009-01-03	0.7183
2009-01-04	0.7187
2009-01-05	0.7188
2009-01-06	0.7271

در این کد ما تنظیمات چندی برای بدست آوردن اطلاعات به روز از وب از دو علامت متفاوت مشخص . این کار از طریق تابع Oanda از خدمات مالی یاهو و نرخ تبدیل دلار - یورو آمریکا از IBM کردیم. با R در قسمت جاری getSymbols()، که اطمینان بخش هر استفاده بعدی از تابع setSymbolLookup() تعیین کننده‌های خاص در فراخوان می‌باشد، در تنظیماتی که ما می‌خواهیم استفاده می‌شود. در این رابطه، دستورالعمل دوم از دو علامتی که ما اطلاعات آن را به شکل خاصی استفاده کردیم قیمت جاری را بدست می‌تواند جهت ذخیره و بارگذاری این loadSymbolLookup() و saveSymbolLookup() می‌آورد. توابع استفاده شود. سودمندی این توابع را برای مثال‌های آتی و بیشتر در R تنظیمات در خلال بخش‌های متفاوت خلال توضیحاتی از کار کردن در پس این توابع سودمند چک کنید.

MySQL 3_2_4- خواندن اطلاعات از دیتابیس

از ذخیره سازی اطلاعات در این مطالعه استفاده می شود. در وب MySQL فرم انتخابی دیگری در دیتابیس برای MySQL است که می تواند دانلود شده و در داخل SQL سایت کتاب فایلی شامل دستورالعمل های به نرخ جاری یک جدول دیتابیس اجرا گردد. اطلاعات استفاده شده و تولید شده از S&P500 بارگذاری می تواند در بخش 3_1 یافت شود. MySQL دیتابیس

بعد از ایجاد یک دیتابیس برای ذخیره کردن ارزش جاری سهام، ما آماده هستیم که دستورالعمل هایی از فایل دانلود شده از سایت کتاب را اجرا کنیم. فرض کنید که این فایل در همان دایرکتوری از جایی که شما را وارد کرده اید باشد و اینکه دیتابیس شما با نام مشابهی از قیمت های جاری ایجاد شده است، شما MySQL را وارد شده و سپس تایپ کنید MySQL می توانید به

```
mysql> use Quotes;  
mysql> source sp500.sql;
```

(فایل دانلود شده از وب سایت کتاب) یک جدول با نام sp500.sql مشتمل بر فایل SQL دستورالعمل های ایجاد خواهد کرد و چندین مدرک الحاقی در این جدول شامل اطلاعات قابل دستیابی برای این مطالعه gspc می توانید تایید کنید که همه چیز MySQL نیز در آن وجود دارد. با اجرای دستورالعمل های زیر در کاراکتر برای اجرا درست است:

```
mysql> show tables;
```

```
+-----+  
| Tables_in_Quotes |  
+-----+  
| gspc |  
+-----+  
1 row in set (0.00 sec)
```

```
mysql> select * from gspc;
```

. اگر شما S&P500 باید رکوردهای زیادی را پرینت کند، برای مثال ارزش جاری SQL آخرین دستور می خواهید محدودیتی برای این خروجی ایجاد کنید، به آسانی می توانید با افزودن محدودیت 10 در پایان دستورالعمل این کار را انجام دهید.

و دیگری بر مبنای ODBC وجود دارد. یکی بر مبنای پروتکل 2R راه اساسی برای ارتباط با دیتابیس در به (DBI (R Special Interest Group on Databases, 2009) واسط عمومی بوسیله پکیج فراهم می کند. (DBMS) همراه پکیج ویژه برای هر سیستم مدیریتی دیتابیس

استفاده کنید، می‌توانید مطمئن باشید که شما قادر به ارتباط با ODBC اگر تصمیم دارید از پروتکل است. از سمت DBMS هستید. این ممکنه شامل نصب چندین درایور در سمت DBMS استفاده از این پروتکل نیاز داشته باشید. RODBNC شما ممکنه به پکیج R

یک سری از توابع واسط دیتابیس را اجرا می‌کند. این توابع از سرور دیتابیس مستقل هستند که DBI پکیج در حقیقت برای ذخیره داده‌ها استفاده می‌شود. استفاده کننده فقط نیاز دارد که نشان دهد وقتی یک ارتباط به دیتابیس برقرار می‌شود کدام میانجی ارتباطی در اولین مرحله استفاده خواهد شد. این بدان معنی است که اگر خود را تغییر دهید، شما تنها به تغییر یک دستورالعمل واحد نیاز خواهید داشت (شما آرزومند DBMS شما را مجزا می‌گرداند). به عبارتی برای دستیابی به این استفاده کننده مستقل DBMS ارتباط با آن هستید که R مختلف مراقبت می‌کند. DBMS همچنین نیاز به نصب پکیج‌هایی دارید که از جزئیات ارتباطی برای هر های اصلی و بزرگ دارد. مخصوصاً، برای ارتباط با یک دیتابیس DBMS بسیاری برای DBMS پکیج‌های ویژه (James and DebRoy, 2009) دارید RMySQL ذخیره شده در بعضی سرورها شما پکیج MySQL

در حال اجرا در ویندوز R3-2-4-1- بارگذاری اطلاعات درون

بر روی همان کامپیوتر MySQL را اجرا می‌کنید، مستقل از اینکه آیا سرور دیتابیس R اگر شما در ویندوز مستقر است یا خیر (مشروط بر اجرا شدن روی سیستم عامل دیگری)، ساده‌ترین راه برای ارتباط با دیتابیس از دارید. RODBNC، شما نیاز به نصب پکیج R است. به عبارت دیگر این پروتکل در ODBC از طریق پروتکل R

باشید ODBC برای اولین بار استفاده از پروتکل MySQL قبل از آنکه قادر به برقراری ارتباط با دیتابیس بر روی سیستم ODBC MySQL چند قدم فراتر ضروری است. برای مثال، شما همچنین به نصب درایور دانلود گردد. این کار تنها MySQL نامیده می‌شود و می‌تواند از طریق سایت myodbc ویندوز نیاز دارید که مورد نیاز است. بعد از نصب این درایور شما می‌توانید MySQL جهت وصل به ODBC برای اولین بار استفاده از مستقر بر روی کامپیوترتان یا هر سیستم دیگری که شما از طریق شبکه محلی MySQL به ODBC اتصال هر اتصال دیتابیزی که شما ایجاد می‌کنید ODBC بدان دسترسی نیاز دارید را ایجاد نمایید. بر طبق پروتکل (این نام برای دسترسی به دیتابیس ODBC بر طبق زبان فنی DSN یک نام دارد (نام منبع اطلاعات، یا بر روی ویندوز کامپیوتر باید برنامه‌ای به نام ODBC استفاده خواهد شد. برای ایجاد یک اتصال R از MySQL ، داشته باشید که از طریق کنترل پنل ویندوز قابل دستیابی است. بعد از اجرای این "منابع اطلاعاتی یک استفاده کننده از منبع اطلاعاتی جدید ایجاد (myodbc) MySQL ODBC برنامه باید با استفاده از درایور کنید که شما فرض می‌شود در گذشته اینستال کرده‌اید. در فرایند ایجاد این تولید، از شما چندین چیز مانند است برای مثال، اگر این یک سرور localhost (اگر کامپیوتر شخصیتان است منظور MySQL آدرس سرور می‌باشد)، نام این دیتابیس که شما می‌خواهید یک کانکشن در آن ایجاد myserver.xpto.pt دور دست است (DSN) (به مثال‌های نقل شده در قبل مراجعه کنید) و نامی که آرزو دارید به این کانکشن بدهید (یعنی پرسیده می‌شود. یک بار که شما این فرایند را تکمیل کردید، که شما تنها برای اولین بار این کار را انجام می‌باشید. R از MySQL می‌دهید، آماده برای اتصال به دیتابیس

را به یک S&P 500 را ایجاد می‌کند و اطلاعات R زیرین یک کانکشن به دیتابیس نقل شده از R کد چارچوب اطلاعاتی بارگذاری می‌کند.

```
> library(RODBC)
> ch <-
odbcConnect("QuotesDSN",uid="myusername",pwd="mypassword")
> allQuotes <- sqlFetch(ch,"gspc")
> GSPC <- xts(allQuotes[,-1],order.by=as.Date(allQuotes[,1]))
> head(GSPC)
```

	Open	High	Low	Close	Volume	AdjClose
1970-01-02	92.06	93.54	91.79	93.00	8050000	93.00
1970-01-05	93.00	94.25	92.53	93.46	11490000	93.46
1970-01-06	93.46	93.81	92.13	92.82	11460000	92.82
1970-01-07	92.82	93.38	91.93	92.63	10010000	92.63
1970-01-08	92.63	93.47	91.99	92.68	10670000	92.68
1970-01-09	92.68	93.25	91.82	92.40	9380000	92.40

```
> odbcClose(ch)
```

DSN، ما یک کانکشن با استفاده از دیتابیس‌مان که در گذشته به عنوان RODBC بعد از بارگذاری پکیج ایجاد می‌کنیم. سپس ما یکی از تابع‌های در دسترس `odbcConnect()` درست کرده بودیم و با استفاده از تابع است که شامل تمام ردیف‌های `sqlFetch()` را برای جستجو در یک جدول استفاده می‌کنید، در این مورد تابع از این چارچوب `xts` جدول و بازده آن‌ها همچون یک فریم اطلاعات می‌باشد. مرحله بعد خلق یک شیء اطلاعات با استفاده از اطلاعات و قیمت‌های جاری است. در نهایت، ما کانکشن را به دیتابیس با تابع `odbcClose()` می‌بندیم.

یک نکته خلاصه در کار کردن به دیتابیس‌های بسیار بزرگ‌تر این است که اگر جستجوی شما یک نتیجه بسیار بزرگ برای مطابقت یافتن با حافظه اصلی کامپیوترتان تولید کند، سپس شما مجبور به استفاده از استراتژی دیگری هستید. اگر آن برای تجزیه و تحلیل شما امکان‌پذیر است، شما می‌توانید برای مدیریت کردن اطلاعات در مقادیر قابل توجه و بزرگ تلاش کنید و این می‌تواند با دستیابی به پارامترهای بزرگتری از توابع مقدور گردد. سایر اهداف می‌تواند در عملکرد عالی و وظیفه مشاهده `sqlFecthMore()` و `sqlFect()` (Adler et al., 2010) محاسبات موازی یافت شود، مثلاً در پکیج

در حال اجرا در لینوکس R 3_2_2-4_2 بارگذاری اطلاعات درون

را از بسته‌های مربوط به یونیکس اجرا می‌کنید آسان‌ترین راه برای ارتباط با دیتابیس R در مواردی که شما ODBC می‌باشد. هنوز پروتکل RMySQL در ملحقات اضافی با پکیج DBI تان احتمالاً از طریق پکیج MySQL شما نیازی به هیچ مرحله مقدماتی مانند RMySQL برای این سیستم‌های عامل قابل دستیابی است. با پکیج ندارید. بعد از اینستال کردن پکیج شما می‌توانید استفاده از آن را همچنانکه با مثال زیر نشان داده‌ایم RODBC آغاز نمایید.

```
> library(DBI)
> library(RMySQL)
> drv <- dbDriver("MySQL")
> ch <- dbConnect(drv, dbname="Quotes", "myusername", "mypassword")
> allQuotes <- dbGetQuery(ch, "select * from gspc")
> GSPC <- xts(allQuotes[, -1], order.by=as.Date(allQuotes[, 1]))
> head(GSPC)
```

	Open	High	Low	Close	Volume	AdjClose
1970-01-02	92.06	93.54	91.79	93.00	8050000	93.00
1970-01-05	93.00	94.25	92.53	93.46	11490000	93.46
1970-01-06	93.46	93.81	92.13	92.82	11460000	92.82
1970-01-07	92.82	93.38	91.93	92.63	10010000	92.63
1970-01-08	92.63	93.47	91.99	92.68	10670000	92.68
1970-01-09	92.68	93.25	91.82	92.40	9380000	92.40

```
> dbDisconnect(ch)
```

```
[1] TRUE
```

```
> dbUnloadDriver(drv)
```

با معنایی `dbConnect()` و `dbDriver()` functions بعد از بارگذاری این پکیج‌ها با استفاده از تابع به دیتابیس SQL به ما برای فرستادن یک سوال `dbGetQuery()` آشکار، اتصال به دیتابیس باز می‌کنیم. تابع `xts` دریافت نتیجه‌ای همچون یک چارچوب اطلاعاتی اجازه می‌دهد. بعد از مکالمات معمولی برای یک شیء می‌بندیم. توابع بیشتر، `dbUnloadDriver()` و `dbDisconnect()` ما کانکشن دیتابیس را برای استفاده از موجود است و DBI شامل توابعی هستند که بخشی از آزمون‌های بزرگ را به دست می‌دهد، همچنین در پکیج ممکنه در اسناد پکیج مورد مشورت قرار گیرد.

استفاده از زیربنای فراهم شده MySQL احتمال دیگر در رابطه با استفاده از اطلاعات در یک دیتابیس `getSymbols()` است که ما در بخش 3_2_3 توضیح دادیم. در نتیجه، تابع `quantmod` بوسیله پکیج

استفاده شود. در زیر دستور العمل ساده‌ای از آن با فرض یک MySQL می‌تواند همچون یک منبع دیتابیس دیتابیس مانند آنچه در بالا شرح دادیم استفاده می‌شود:

```
> setSymbolLookup(GSPC=list(name='gspc',src='mysql',  
+ db.fields=c('Index','Open','High','Low','Close','Volume','AdjClose'),  
+ user='xpto',password='ypto',dbname='Quotes'))  
> getSymbols('GSPC')
```

[1] "GSPC"

3-3- تعریف وظایف پیش‌بینی

داشته باشیم در S&p500 عموماً، هدف ما این است که پیش‌بینی خوبی برای قیمت‌های آتی شاخص نتیجه آن سفارش سودآور می‌تواند به موقع مورد استفاده قرار گیرد. این هدف کلی باید به ما اجازه دهد به آسانی آنچه را برای پیش‌بینی مدل‌مان نیاز داریم تعریف کند - این امر باید برای پیش‌بینی ارزش آتی سری زمانی قیمت‌ها دسته‌بندی شود. اگرچه، فهمیدن آن مطابق این وظیفه ساده آسان است، ما فوراً با چندین سوال مواجه می‌شویم، برای مثال (1) کدامیک از قیمت‌های روزانه مد نظر است؟ یا (2) برای چه زمانی در آینده باید محسوب شود؟ جواب دادن به این سوالات ممکنه آسان نباشد و معمولاً به اینکه چگونه پیش‌بینی‌ها برای تولید سفارشات تجاری مورد استفاده قرار می‌گیرد، بستگی دارد.

3-3-1- چه چیزی را پیش‌بینی کنیم؟

استراتژی‌های مبادلاتی که ما در بخش 3-5 توضیح خواهیم داد فرض می‌کند که ما یک پیش‌بینی از گرایش نمرات در چند روز آینده بدست آوردیم. بر اساس این پیش‌بینی ما دستوراتی که اگر گرایش اعداد در آینده تایید گردد سودآور خواهد بود را جایگزین خواهیم کرد.

باشد، ما این را در اصطلاح تجارت به عنوان $p\%$ اجازه دهید فرض کنیم که اگر قیمت‌ها خیلی بیش از مبادله‌ای با ارزش بررسی می‌کنیم (برای مثال، پوشش دادن هزینه مبادلات). در این نوشته، ما مدل پیش‌بینی‌مان روز آینده می‌خواهیم. لطفاً توجه داشته باشید که k برای پیشگویی آنچه این حاشیه سود قابل دستیابی است در روی ما حقیقتاً می‌توانیم قیمت‌ها را در بالا و پایین این درصد مشاهده کنیم. این بدین معنی k در خلال این ممکنه بهترین فکر نباشد. در نتیجه، آنچه $t+k$ است که پیش‌بینی کردن یک قیمت ویژه برای زمان آتی خاصی روز آتی است و این بوسیله یک قیمت خاصی در زمان k ما می‌خواهیم داشتن پیش‌بینی از قیمت پویای کلی در ممکنه بیان کننده یک اختلاف خیلی کوچک‌تر از $t+k$ خاص بدست نمی‌آید. مثلاً، قیمت بسته شدن در زمان $p\%$ باشد اما همین امر ممکن است در فرایند محاسبه قیمت دوره، در نشان دادن اختلافی بسیار بیشتر از $p\%$ با اهمیت و ارجح باشد. بنابراین آنچه ما به عنوان نتیجه می‌خواهیم برای داشتن یک $t+k$ t در پنجره روز آینده است. k پیش‌بینی خوب از گرایش کلی قیمت در

ما یک متغیر که با اطلاعات قیمت جاری محاسبه خواهد شد را شرح خواهیم داد، که می‌تواند همچون یک روز آتی مشاهده شود. ارزش این شاخص باید با اطمینان از اینکه ما به آن k شاخص (یک بها) از گرایش در روز آتی قابل دستیابی است، مرتبط باشد. در این مرحله این مهمه که توجه داشته k داریم و در p حاشیه هدف را مورد توجه داریم منظورمان زیر یا بالای قیمت جاری است. هدف این $p\%$ باشیم که وقتی ما یک انحراف از است که انحراف مثبت ما را به خرید هدایت می‌کند در حالیکه انحراف منفی فعالیت‌های فروش را برمی‌انگیزاند. شاخصی که پیشنهاد می‌کنیم حاصل گرایشی همچون یک بهای واحد است، مثبت برای گرایش‌ها به سمت بالا و منفی برای گرایش‌ها به سمت پایین قیمت‌ها. اجازه دهید قیمت میانگین روزانه را بوسیله تقریب زیر بدست آوریم:

$$\bar{P}_i = \frac{C_i + H_i + L_i}{3}$$

V_i است. اجازه دهید i بسته می‌شود به ترتیب بالا و پایین قیمت جاری برای روز L_i و H_i و C_i جایگاه بسته می‌شود (اغلب بازگشت‌های k به قیمت میانگین روزهای پیامد k مجموعه‌ای از درصد انحرافات امروز حساب نامیده می‌شود):

$$V_i = \left\{ \frac{\bar{P}_{i+j} - C_i}{C_i} \right\}_{j=1}^k$$

ما است: $p\%$ تغییرات شاخص ما مجموع کلی انحرافات است که ارزش مستقل بالاتر از حاشیه هدف

$$T_i = \sum_v \{v \in V_i : v > p\% \vee v < -p\%\}$$

روز است که چندین روز با میانگین قیمت‌های روزانه به طور k برای یک دوره واحد t دیدگاه کلی تغییرات بدین معنی است که آنجا چندین میانگین قیمت t آشکاری بیش از تغییرات هدف دارد. ارزش مثبت بالای است که امروز بسته می‌شود. چنین موقعیتی شاخص خوبی از p روزانه وجود دارد که بیش از بالاتری درصد فرصت‌های بالقوه برای انتشار یک دستور خرید است، همچنانکه ما توقعات مناسبی از اینکه قیمت‌ها افزایش قیمت‌های داده شده احتمالاً T خواهد یافت. به عبارت دیگر، ارزش منفی بالای فعالیت‌های فروش پیشنهادی کاهش خواهد یافت. ارزش نزدیک صفر می‌تواند به وسیله دوره‌هایی با قیمت‌های تخت یا با انحرافات مثبت و منفی تداخلی که همدیگر را خنثی می‌کند.

تابع زیر این شاخص ساده را اجرا می‌کند:

```
> T.ind <- function(quotes, tgt.margin = 0.025, n.days = 10) {
+ v <- apply(HLC(quotes), 1, mean)
+ r <- matrix(NA, ncol = n.days, nrow = NROW(quotes))
+ for (x in 1:n.days) r[, x] <- Next(Delt(v, k = x), x)
+ x <- apply(r, 1, function(x) sum(x[x > tgt.margin | x <
```

```

+ -tgt.margin]))
+ if (is.xts(quotes))
+ xts(x, time(quotes))
+ else x
+ }

```

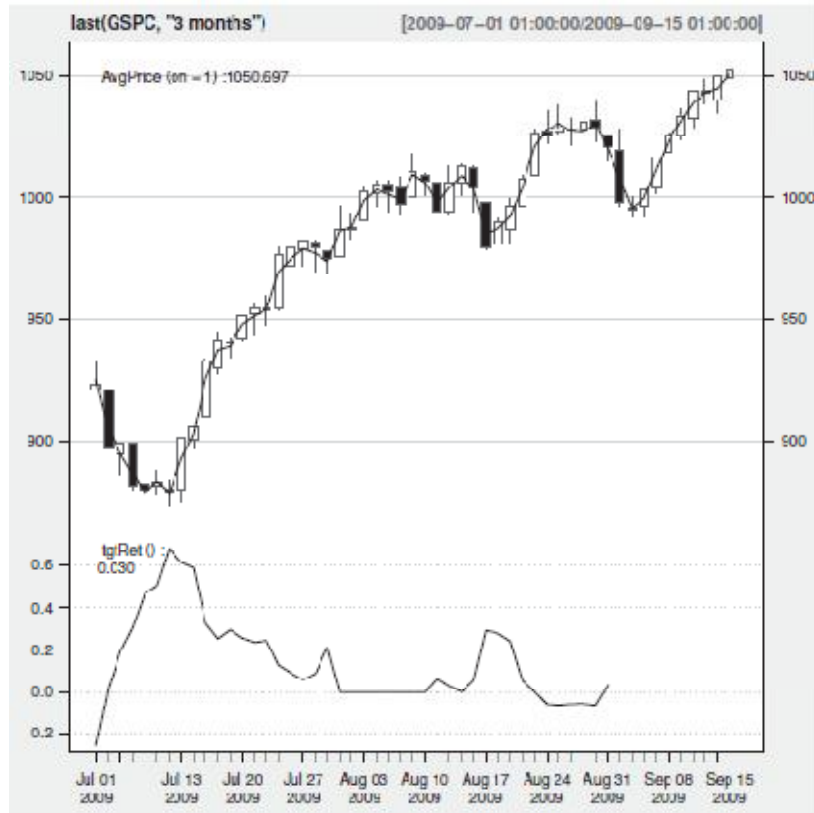
قیمت‌های بالا، HLC() تابع با بدست آوردن محاسبه قیمت میانگین برطبق معادله 2-3 آغاز می‌گردد. تابع روز آتی را با توجه به قیمت n پایین و بسته شدن را از شیء قیمت جاری استخراج می‌کند. ما سپس بازده به جابجایی ارزش سری زمانی در زمان (هر دو زمان آینده و Next() بسته جاری به دست می‌آوریم. تابع می‌تواند برای محاسبه درصد یا بازده طولانی یک سری از قیمت‌ها استفاده (Delt() گذشته) اجازه می‌دهد. تابع جمع می‌کند بازگشت‌های قطعی بسیاری را جمع می‌کند که بازگشت‌های بالا T.ind() شود. در نهایت، تابع حاشیه انحراف هدفی است که بوسیله مقدار پیش‌فرض 2/5 درصد تنظیم می‌شود.

ما می‌توانیم نظریه بهتری از رفتار این شاخص در نمودار 1-3 که با کدهای ذیل تولید شده است را بدست آوریم:

```

> candleChart(last(GSPC, "3 months"), theme = "white", TA = NULL)
> avgPrice <- function(p) apply(HLC(p), 1, mean)
> addAvgPrice <- newTA(FUN = avgPrice, col = 1, legend = "AvgPrice")
> addT.ind <- newTA(FUN = T.ind, col = "red", legend = "tgtRet")
> addAvgPrice(on = 1)
> addT.ind()

```



در سه ماه گذشته و شاخص S&P500 نمودار 1-3:

از قیمت جاری سهام رسم می‌کند. این نمودارها `candlestick` نمودارهای `candleChart()` تابع قیمت‌های جاری روزانه را بوسیله جعبه رنگی و نمودارهای میله‌ای عمودی ارائه می‌دهد. نمودار میله‌ای قیمت‌های بالا و پایین روز را ارائه می‌دهد در حالیکه جعبه دامنه باز و بسته بودن آن را نشان می‌دهد. رنگ جعبه نشان بر اینست که بالای جعبه قیمت باز یا بسته است که اگر قیمت‌ها کاهش یابد (سیاه در نمودار 1-3)، در میان بخش‌های روزانه افزایش یابد. R) یا (سفید در نمودار ما، سبز در بخش R نارنجی در یک بخش تعاملی `candlesticks` را به دو شاخص نمودار اضافه کردیم: قیمت میانگین (در همان نمودار مانند `candlestick` ما می‌تواند برای خلق توابع جدید ترسیم جهت برای شاخص‌هایی که ما آرزو `newTA()` ما (زیر). تابع T و شاخص قرار دهیم می‌باشد. ارزش بازده این تابع یک تابع ترسیم است! این بدین `candlestick` داریم در نمودارهای R اضافه می‌شود می‌تواند همچون هر تابع دیگری از `addAvgPrice` و `T.ind` معنی است که اشیایی که به نامیده شود. این امر بر روی دو دستورالعمل آخر انجام می‌شود. هر کدام از آن‌ها یک شاخص به نمودار اولیه با پارامتر تنظیم شده برای 1 `addAvgPrice()` اضافه می‌کند. تابع `candleChart()` تولید شده بوسیله تابع نامگذاری می‌شود که بدین معنی است که شاخص روی اولین پنجره نمودار ترسیم خواهد شد؛ که آنجایی با این استدلال نامگذاری نمی‌شود، به یک نمودار جدید `addT.ind()` است. تابع `candlestick` می‌باشد که

هدایت می‌کند. دادن مقیاس کاملاً متفاوتی از ارزش چیزی است که در مورد شاخص مان $candlesticks$ زیر زمانی که دوره بعدی از انحرافات T مفهوم می‌یابد. همچنانکه می‌توانید در نمودار 1-3 مشاهده کنید، شاخص مثبت وجود دارد به بالاترین ارزش و بها دست می‌یابد. واضح است جهت دستیابی به ارزش شاخص برای زمان این حرکت‌ها را پیش‌بینی T لازم است به ارزش جاری برای 10 روز بعدی را داشته باشیم بنابراین نمی‌گوییم که می‌کند. این هدف شاخص نیست. این هدفی برای خلاصه سازی مشاهدات رفتار آتی از قیمت‌ها در یک ارزش واحد است و این رفتار را پیش‌بینی نمی‌کند! در دیدگاه ما به این مسئله فرض خواهیم کرد فعالیت‌های تجاری روز آتی بستگی دارد وابسته است. k به آنچه انتظارات ما در رابطه با ارزیابی قیمت‌ها در t صحیح در زمان مان را شرح خواهیم داد. نشانه مبادله صحیح در t علاوه، ما این فرضیه تکاملی آتی از قیمت‌ها بوسیله شاخص بالاتر از یک آستانه معین باشد خریداری خواهد شد و اگر نمره زیر آستانه دیگری باشد t اگر نمره t زمان فروخته خواهد شد. در تمام موارد دیگر، نشانه صحیح هیچ کاری انجام نخواهد داد (مانند نگهداری). به طور باشیم. راجع به اطلاعات تاریخی ما علامت t خلاصه، ما می‌خواهیم قادر به پیش‌بینی علامت صحیح برای زمان مربوطه و استفاده از روش آستانه مانند خلاصه بالا، پر خواهیم کرد. T صحیح برای هر روز را با محاسبه نمره

3-2-3- کدام پیش‌بینی کننده‌ها؟

روز آتی را خلاصه می‌کند. هدف k تعریف می‌کنیم که رفتار سری‌های زمانی قیمت در T ما یک شاخص داده‌کاویمان این رفتار را پیش‌بینی خواهد کرد. فرضیات اصلی پشت تلاش برای پیش‌بینی رفتار آتی بازارهای مالی این است که ممکن است بوسیله مشاهده رفتار گذشته بازار انجام شود. صراحتاً، ما فرض می‌کنیم که اگر دنبال می‌شد و اگر آن زنجیره علت به طور مکرر f بوسیله رفتاری دیگری همچون p در گذشته یک رفتار خاص اتفاق می‌افتاد، پس این قابل پذیرش است که فرض کنیم این امر دوباره در آینده اتفاق خواهد افتاد و در نتیجه بعدی مشاهده خواهیم کرد را پیش‌بینی کنیم. ما بطور f را الان مشاهده کنیم می‌توانیم آنچه در p اگر ما مان برآورد می‌کنیم. حالا باید تصمیم بگیریم چگونه الگوی قیمت‌های T را به وسیله شاخص f تقریبی رفتار آتی در توضیحات بالا). به جای استفاده مجدد از یک شاخص واحد برای توضیح این p اخیر را شرح خواهیم داد (فعالیت اخیر، با تلاش برای تصرف اموال متفاوتی از سری‌های زمانی قیمت برای تسهیل وظیفه پیش‌بینی، چندین شاخص را استفاده خواهیم کرد.

آسان‌ترین نوع اطلاعات ما می‌توانیم برای توصیف گذشته استفاده کنیم قیمت‌های مشاهده شده اخیر است. به طور غیررسمی، آن نوعی از روشی است که در چندین مدل سری‌های زمانی استاندارد قابل دستیابی است. این مدل‌های خطی مشی توسعه است که ارتباط بین ارزش‌های آتی سری‌های زمانی و پنجره‌ای به گذشته این سری‌های زمانی را توصیف می‌کند. ما برای غنی سازی توصیف‌مان از فعالیت‌های اخیر q مشاهدات سری‌های زمان با افزودن آینده‌های بیشتری به این پنجره از قیمت‌های جاری، تلاش خواهیم کرد.

شاخص‌های فنی خلاصه‌های عددی است که بعضی دارایی‌های قیمت این سری‌های زمان را منعکس می‌کند. برخلاف دیتابیس آن‌ها که همچون ابزاری برای تصمیم‌گیری اینکه چه زمانی مبادله انجام شود، استفاده

می‌شود آن‌ها همچنان می‌توانند خلاصه جذابی از پویایی قیمت سری‌های زمان فراهم کنند. مقدار شاخص‌های ما می‌توانیم مثال ساده خوبی پیدا کنیم. R، در TTR فنی در دسترس می‌تواند طاقت‌فرسا باشد. با تشکر از پکیج شاخص‌ها معمولا سعی می‌کند بعضی دارایی‌ها از سری‌های قیمت‌ها، مانند اینکه آیا آن‌ها تغییرات زیادی دارند یا نه و یا دنبال کردن بعضی تمایلات خاص و غیره را تبدیل کنیم. برای غالب شدن به این مشکل، ما جستجوی فراگیری برای شاخص‌هایی که برای وظیفه ما مناسب‌تر است را انجام نخواهیم داد. نه تنها برای این برنامه ویژه بلکه هنوز این یک سوال تحقیق مربوط و مناسب است. معمولا این به عنوان مشکل انتخاب شده آتی شناخته می‌شود و می‌تواند به طور غیررسمی به عنوان وظیفه پیدا کردن مناسب‌ترین زیرمجموعه از متغیرهای ورودی در دسترس برای یک مدل وظیفه تعریف شود. دستیابی‌های موجود برای این مسئله می‌تواند معمولا در دو گروه مطرح گردد: 1) فیلترهای آتی و 2) پوشش‌های آتی. اولی استقلال از ابزار مدل‌سازی است که بعد از فاز انتخابی بعدی استفاده خواهد شد. آن‌ها بطور اساسی سعی می‌کنند بعضی دارایی‌های آماری از آینده (برای مثال، همبستگی) را جهت انتخاب گروه نهایی از آینده استفاده کنند. دستیابی‌های پوششی شامل ابزار مدل‌سازی در فرایند انتخاب است. آن‌ها جایی که در هر قدم یک گروه داوطلب از آینده سعی دارد با ابزار مدل‌سازی کار کند، یک فرایند جستجوی تکراری را انجام می‌دهند و نتایج مربوطه ثبت شده است. بر اساس این نتایج، گروه‌های آزمایشی جدید با استفاده کردن از بعضی تحقیقات عملیاتی تولید می‌شوند، و فرایند تکرار می‌شود تا اینکه بعضی معیارهای همگرایی ملاقات می‌شوند که گروه نهایی را تعریف خواهد کرد.

ما یک دیدگاه ساده جهت انتخاب آینده برای شامل شدن در مدل‌مان استفاده خواهیم کرد. آن دیدگاه این است که این فرایند را با یک مثال محسوس شرح دهیم و نه اینکه بهترین راه حل ممکن برای این مشکل را پیدا کنیم که به زمان منابع محاسباتی دیگری نیاز دارد. ما یک گروه اولیه از خصایص تعریف خواهیم کرد و سپس یک تکنیک برای تخمین زدن اهمیت هر کدام از این خصلت‌ها استفاده می‌کنیم. بر اساس این تخمین‌ها ما بیشتر خصایص مربوط را انتخاب خواهیم کرد. ما تجزیه و تحلیل‌مان را بر روی قیمت جاری بسته شد متمرکز می‌کنیم همچنانکه تصمیمات خرید و فروشمان در پایان هر دوره روزانه ساخته خواهد شد. اولین گروه خصایص (محاسباتی) یا درصد h بوسیله چندین بازده گذشته از قیمت بسته شدن شکل خواهد گرفت. بازده روزهای انحرافات می‌تواند بدین شکل محاسبه گردد:

$$R_{i-h} = \frac{C_i - C_{i-h}}{C_{i-h}}$$

است. در گروه خصلت‌های داوطلبین، ده تا از این بازده‌ها را با i قیمت بسته شدن در بخش C_i در حالی که در دسترس در پکیج یک گروه نمایشگر از شاخص‌های TTR از 1 تا 10 محاسبه می‌کنیم. پس از آن h تغییرات فنی را انتخاب می‌کنیم، برای مثال، دامنه صحت میانگین که یک شاخص فرارایت این سری‌ها است؛ شاخص (ADX) که یک شاخص مقدار حرکت است؛ شاخص حرکت هدایتی والس ویلدر (SMI) اندازه‌های تصادفی شاخص آر‌ن که سعی دارد گرایش‌ات آغازین را تعریف کند؛ شاخص بولینگر بندز که فرارایت را فراتر از یک دوره که به قسمت بسته شدن جهت دامنه مبادلاتی (CLV) زمانی می‌سنجد؛ فرارایت چایکین؛ ارزش مکان بسته

؛ (MFI)؛ شاخص گردش نقدینگی MACD؛ نوسانگر (EMV) خود، وابسته است؛ ارزش حرکتی آرمز ایز ایستادن و توقف سهمی وار؛ و شاخص فراریت. جزئیات و منابع بیشتر برای اینها و دیگر شاخصها می تواند در اجرا می کنند، یافت می شود. بیشتر این شاخصها TTR صفحه کمک های مربوطه از توابعی که آنها را در پکیج چندین بهایی تولید می کند که با همدیگر برای تصمیم سازی های تجاری استفاده می شود. همانطور که در گذشته خاطر نشان شد، برای استفاده از این شاخصها جهت مبادلات برنامه ای نداریم. همینطور، بعضی را جهت بدست آوردن یک بهای واحد برای هر کدام بکار برده شد. TTR فرایندهای گذشته از خروجی توابع توابع زیر این فرایند را اجرا می کند:

```
> myATR <- function(x) ATR(HLC(x))[, "atr"]
> mySMI <- function(x) SMI(HLC(x))[, "SMI"]
> myADX <- function(x) ADX(HLC(x))[, "ADX"]
> myAroon <- function(x) aroon(x[, c("High", "Low")])$oscillator
> myBB <- function(x) BBands(HLC(x))[, "pctB"]
> myChaikinVol <- function(x) Delt(chaikinVolatility(x[, c("High",
+   "Low")]))[, 1]
> myCLV <- function(x) EMA(CLV(HLC(x)))[, 1]
> myEMV <- function(x) EMV(x[, c("High", "Low")], x[, "Volume"])[,
+   2]
> myMACD <- function(x) MACD(CI(x))[, 2]
> myMFI <- function(x) MFI(x[, c("High", "Low", "Close")],
+   x[, "Volume"])
> mySAR <- function(x) SAR(x[, c("High", "Close")])[, 1]
> myVolat <- function(x) volatility(OHLC(x), calc = "garman")[,
+   1]
```

T تغییراتی که از پیش بینی های گروه اولیه مان توصیف کردیم برای وظیفه پیش بینی ارزش آتی شاخص بود. ما جهت کاهش این گروه از 22 متغیر استفاده کردن از یک در روش انتخاب ویژگی تلاش خواهیم کرد. در بخش 2-7 برای بدست آوردن پیش بینی هایی از رخداد خزه استفاده (Beriman, 2001) جنگل تصادفی شد. جنگل تصادفی همچنین می تواند برای تخمین زدن اهمیت متغیرهای شامل شده در وظیفه پیش بینی استفاده گردد. بطور غیررسمی، اگر ما هر متغیر در بازده را حذف کنیم این تقاضا می تواند بوسیله محاسبه افزایش درصد در خطاهای تصادفی پیش بینی محاسبه گردد. در شیوه ای معین این امر نظریه فیلترهای پوشاننده را همانند سازی می کند در نتیجه این شامل یک ابزار مدل سازی در فرایند انتخاب ویژگی ها را شامل می شود. اگرچه این یک فرایند جستجوی تکراری نیست و بعلاوه ما از سایر مدل های پیش بینی برای پیش بینی استفاده خواهیم کرد، این بدین معنی است که گروه متغیرهای انتخاب شده بوسیله این فرایند برای سایر مدل ها

بهینه نمی‌شود و بدینصورت این روش بیشتر شبیه یک فیلتر دستیابی استفاده می‌شود. در برداشت ما از این برنامه، ما اطلاعات در دسترس را به دو بخش مجزا تقسیم می‌کنیم: 1) یکی برای ایجاد سیستم مبادلاتی استفاده می‌شود و 2) باقی جهت آزمایش کردن آن بکار می‌رود. گروه اول بوسیله اولین اطلاعات 30 سال شکل داده خواهد شد. باقی اطلاعات را (حدود 9 سال) برای تست نهایی از S&P500 قیمت‌های جاری از سیستم تجاری مان رها خواهیم کرد. در این ارتباط، باید تست‌های نهایی از این فرایند انتخاب ویژگی جهت مطمئن شدن از بی‌غرض بودن نتایج را رها کنیم. با استفاده از اطلاعات در دسترس برای تمرین ما ابتدا یک پیش‌بینی تصادفی درست می‌کنیم:

```
> data(GSPC)
> library(randomForest)
> data.model <- specifyModel(T.ind(GSPC) ~ Delt(CI(GSPC),k=1:10) +
+   myATR(GSPC) + mySMI(GSPC) + myADX(GSPC) +
+   myAroon(GSPC) +
+   myBB(GSPC) + myChaikinVol(GSPC) + myCLV(GSPC) +
+   CMO(CI(GSPC)) + EMA(Delt(CI(GSPC))) + myEMV(GSPC) +
+   myVolat(GSPC) + myMACD(GSPC) + myMFI(GSPC) +
+   RSI(CI(GSPC)) +
+   mySAR(GSPC) + runMean(CI(GSPC)) + runSD(CI(GSPC)))
> set.seed(1234)
> rf <- buildModel(data.model,method='randomForest',
+   training.per=c(start(GSPC),index(GSPC["1999-12-31"])),
+   ntree=50, importance=T)
```

کدهای داده شده در بالا بوسیله علامت‌گذاری آغاز می‌شود و اطلاعات بدست آمده جهت مدل‌سازی با ایجاد می‌کند که quantmod استفاده خواهد شد. این تابع یک شیء `specifyModel()` استفاده از تابع شامل ویژگی خاصی از مدل انتزاعی معینی می‌باشد (یا فرمول توضیح داده شده است). این ویژگی ممکن است به اطلاعات آمده از انواعی از منابع متفاوت که بعضی ممکنه حتی اخیراً در حافظه کامپیوتر نباشد، اشاره کند. برای دستیابی به اطلاعات ضروری مراقبت می‌کند. این مطلب `getSymbols()` تابع از این موارد با استفاده از در شکلی بسیار قابل حصول از ویژگی و بدست آوردن اطلاعات ضروری جهت مراحل مدل‌سازی بعدی نتیجه می‌دهد. بعلاوه، برای علامت‌هایی که در وب منبعی دارد شما می‌توانید بعداً از هدف بدست آمده ، جهت بدست آوردن `getModelData()` در مسئله ما) همچون یک متغیر مستقل برای تابع `data.model`) یک بازیابی هدف مشتمل بر هر قیمت جاری جدید که ممکنه در همان زمان در دسترس باشد، استفاده کنید. از طرف دیگر، این کاملاً متقاعد کننده است اگر شما بخواهید یک سیستم تجاری که باید با اطلاعات جدیدی از ویژگی مدل نتیجه داده را استفاده `buildModel()` قیمت‌های جاری مرتب به روز شود نگهداری کنید. تابع

، شما می‌توانید اطلاعات که `trainin.per` می‌کند و شامل یک مدل با اطلاعات مشابه است. از طریق، پارامتر باید برای بدست آوردن مدل استفاده شود را مشخص سازید (ما برای 30 سال اول استفاده می‌کنیم). این تابع اخیراً شامل پوشاننده‌هایی برای ابزارهای مدل‌سازی چندی در میان آن‌ها که پیش‌بینی تصادفی هستند، استفاده نکنید، ممکنه `buldModel()` باشد. در جایی که شما آرزو دارید از یک مدل بررسی شده بوسیله بدست آورید و آن را در تابع مدل‌سازی مورد علاقه‌تان همچنانکه `modelData()` اطلاعات را با استفاده از تابع در مثال گویای زیر نشان داده شده است، استفاده کنید:

```
> ex.model <- specifyModel(T.ind(IBM) ~ Delt(CI(IBM), k = 1:3))
> data <- modelData(ex.model, data.window = c("2009-01-01",
+      "2009-08-10"))
```

ی استاندارد است که می‌تواند به سادگی در یک ماتریک یا zoo هدف اطلاعات شامل شده یک هدف چارچوب اطلاعات، بوسیله استفاده همچون یک پارامتر از هر تابع مدل‌سازی مانند مثال گویای ساختگی زیر طرح‌ریزی و قالب‌بندی شود:

```
> m <- myFavouriteModellingTool(ex.model@model.formula,
+      as.data.frame(data))
```

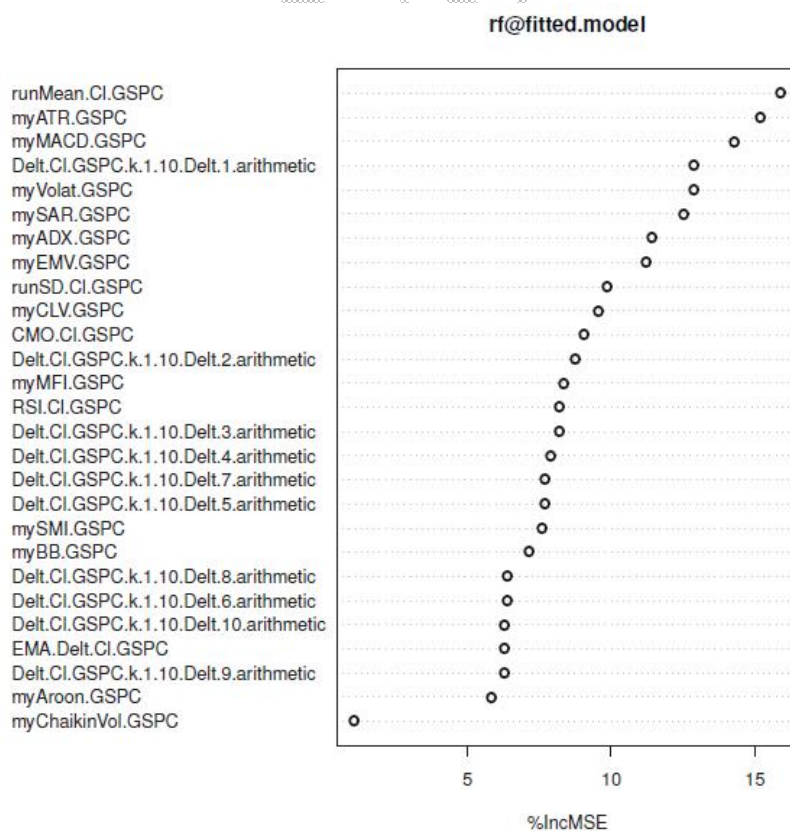
توجه داشته باشید چگونه فرمول مدل را نشان دادیم. فرمول واقعی دقیقاً شبیه آنچه در در متغیر مستقل از فراهم شده است نمی‌باشد. این فرمول اخیر برای رفتن و آوردن اطلاعات استفاده شده `specifyModel()` تابع تولید می‌کند `specifyModel()` است، اما فرمول واقعی باید برای هر کدام از ستون‌ها و اسامی مربوطه که تابع مشتمل `quantmod` از شیء تولید شده بوسیله تابع `model.formula` استفاده گردد. این اطلاعات بر شیار می‌شود. توجه داشته باشید که در این مثال ساختگی کوچک ما به یک وسیله که اطلاعات را بر روی یک ریبون ، برای آنهایی که اخیراً در اطلاعات حافظه نداریم، اشاره کردیم. تابع `(IBM)` باریک کاغذی پرینت می‌کند از آن بوسیله واکنشی خاموشی از اطلاعات قیمت جاری بدست آمده از وب با استفاده از تابع `specifyModel()` مراقبت می‌کند. تمام اینها در یک شکل نامرئی برای استفاده کننده انجام می‌شود و شما `getSumbols()` ممکن است حتی علامت‌هایی در ویژگی مدلتان که مشتمل بر منابع متفاوتی می‌باشد (برای نمونه به مثال‌های مراجعه کنید) را به حساب آورید. `setSymbolLookup()` بخش 3-2-3 با استفاده از تابع

در بازده به مشکل انتخاب خصیصه‌مان توجه داشته باشید که ما پارامتر اهمیت = صحیح را به حساب آوردیم بنابراین آن پیش‌بینی‌های تصادفی از اهمیت متغیر را تخمین می‌زند. برای مشکلات رگرسیونی، اجرای از پیش‌بینی‌های تصادفی اهمیت متغیر را با دو منبع متناوب تخمین می‌زند. اولی افزایش درصد در خطای R پیش‌بینی است اگر ما هر متغیر در بازده را حذف کنیم. این امر وقتی که هر متغیری حذف شود با محاسبه

افزایش در میانگین مربع خطای هر درخت در یک نمونه خارج از کیف اندازه‌گیری می‌گردد. این افزایش میانگین فراتر از همه درخت‌ها در جنگل است و با خطای استاندارد بهنجار شده است. این رتبه دوم به وسیله کاهش در منحنی ناخالصی (آلودگی) که با هر متغیری قابل محاسبه است، باید انجام شود که باز هم میانگین بیشتر از تمام درخت‌ها است. ما از رتبه اول همچنانکه آن همانی است که در کاغذ اصلی پیش‌بینی‌های تصادفی . پس از بدست آوردن مدل، ما می‌توانیم اهمیت متغیرها را (Breiman, 2001) اشاره شد استفاده خواهیم کرد . مانند زیر چک کنیم:

```
> varImpPlot(rf@fitted.model, type = 1)
```

پیش‌بینی (varImpPlot) نتیجه فراخوانی این تابع در نمودار 2-3 داده شده است. متغیرهای مستقل تابع تصادفی از جنگل و رتبه‌هایی است که آرزو داریم ترسیم شود (اگر هر دو در ترسیم حذف شوند). تابع کلی quantmod از شی 2 2 (fitted model) مدل بدست آمده را همچون شیاری (buildModel) بازگشت‌های که مانند یک نتیجه تولید می‌شود را فراهم می‌آورد.



نمودار 2-3: اهمیت متغیر برطبق پیش‌بینی تصادفی

در این مرحله لازم است برای یک آستانه اهمیت رتبه برای تنها یک زیرمجموعه از خصایص انتخاب کنیم. با نگاه کردن به نتایج روی نمودار و دادن آن، این مثال ساده از مفهوم استفاده از پیش‌بینی تصادفی برای خصایص انتخاب شده است، ما از بهای 10 به عنوان آستانه استفاده خواهیم کرد.

```
> imp <- importance(rf@fitted.model, type = 1)
> rownames(imp)[which(imp > 10)]
```

```
[1] "Delt.Cl.GSPC.k.1.10.Delt.1.arithmetic"
[2] "myATR.GSPC"
[3] "myADX.GSPC"
[4] "myEMV.GSPC"
[5] "myVolat.GSPC"
[6] "myMACD.GSPC"
[7] "mySAR.GSPC"
[8] "runMean.Cl.GSPC"
```

نمرات محسوسی را برای هر متغیر فراهم می‌کند (در این مورد اولین رتبه است) که `importance()` تا به ما سپس برای بدست آوردن نام‌های متغیرهایی که در مدل آزمایشی مان استفاده خواهیم کرد با آستانه مان فیلتر می‌کنیم. در استفاده کردن این اطلاعات ما می‌توانیم گروه اطلاعات نهایی مان را مانند زیر بدست آوریم:

```
> data.model <- specifyModel(T.ind(GSPC) ~ Delt(Cl(GSPC), k = 1) +
+   myATR(GSPC) + myADX(GSPC) + myEMV(GSPC) +
+   myVolat(GSPC) +
+   myMACD(GSPC) + mySAR(GSPC) + runMean(Cl(GSPC)))
```

3-3-3 وظایف پیش‌بینی

شامل اطلاعاتی که قصد داریم برای استفاده با `quantmod (data.model)` در بخش قبل هدف دارد و `T` مدل‌های پیش‌بینی‌مان استفاده کنیم، را بدست آوردیم. این اطلاعات همچون یک هدف ارزش شاخص مانند پیش‌بینی کننده‌های یک سری دیگر از متغیرها که از فرایند انتخاب یک خصیصه برآمده، نتیجه می‌شود. می‌باشد. چگونه `T` در بخش 3-3-1 مشاهده کردیم که هدف واقعی ما پیش‌بینی صحیح یک مبادله در هر زمان می‌توانیم با استفاده از اطلاعات داده شده که در بخش‌های قبلی تولید کردیم آن کار را انجام دهیم؟ ما دو مسیر برای فراهم کردن پیش‌بینی‌هایی برای تجارت واحد صحیح را کاوش خواهیم کرد.

مانند یک متغیر هدف است و برای فراهم کردن مدلهایی که این ارزش را با `T` اولین چاره استفاده از ارزش استفاده از اطلاعات پیش‌بینی کننده‌ها پیش‌بینی کند، تلاش می‌کند. این یک وظیفه رگرسیون چندگانه است که

شبيه به آن است كه در فصل قبلى مورد بررسى قرار داديم. اگر ما اين مسير را دنبال كنيم، سپس قادر به ترجمه پيش بينى هاى مدلمان به علامت هاى مبادلاتى خواهيم بود. اين به تصميم گيرى كردن فراتر از آستانه كه به هر كدام از سه فعاليت مبادلاتى ممكن هدايت خواهد كرد، معنى مى دهد. ما اين آ ارزش پيش بينى شده تبديل را با استفاده از ارزش هاى زير اجرا خواهيم كرد:

$$signal = \begin{cases} \text{sell} & \text{if } T < -0.1 \\ \text{hold} & \text{if } -0.1 < T < 0.1 \\ \text{buy} & \text{if } T > 0.1 \end{cases}$$

اين بخش از ارزش هاى 0/1 و -0/1- بكلى مبتكرانه است و ما همچنين مى توانيم ديگر آستانه ها را استفاده حداقل كنيم. هنوز اين ارزش ها به معنى اين است كه در خلال دوره 10 روزه استفاده شده براى توليد ارزش 4 ميانيگين قيمت روزانه كه بالاي قيمت بسته جارى 2/5% هستند (0/1 = 0/025 * 4). اگر براى استفاده از ساير ارزش ها تصميم بگيريد، بايد در نظر داشته باشيد كه ارزش هاى مستقل بالا علامت هاى كمترى را موجب مى شود در حاليكه ارزش هاى خيلى كمى ممكن است ما را به تجارت بر اساس انحرافات بسيار كوچك بازار كه در پكيچ كتاب قابل دستيابى ()trading.signals هدايت كند، بنابر اين ريسك بزرگترى ايجاد مى شود. تابع براى فعاليت b و h و s به يك فاكتر با سه ارزش ممكن T است مى تواند اين دگرگونى را از ارزش هاى عددى فروش، نگهدارى و خريد مربوطه انجام دهد.

ما وظيفه دوم جاگزين پيش بينى را از پيش بينى مستقيم علامت ها در نظر مى گيريم. اين وسيله به عنوان استفاده مى شود. چگونه ما اين علامت هاى درست را بدست d يك متغير هدف علامت درست براى روز استفاده مى كنيم و همان آستانه استفاده شده در معادله 3-6. براى اطلاعات T مى آوريم؟ دوباره از شاخص با استفاده از 10 روز بعدى بدست مى آوريم و از T تاريخى در دسترس ما نشانه هر روز را بوسيله محاسبه ارزش آستانه ها در معادله 3-6 براى تصميم گيرى علامت استفاده مى شود. متغير هدف در اين وظيفه ثانويه اسمى است. اين نوع از مسائل پيش بينى به عنوان وظيفه طبقه بندى شناخته مى شود. فرق اصلى بين طبقه بندى و وظيفه رگرسيون نوع متغير هدف است. وظيفه رگرسيون يك تعدادى متغير هدف دارد (براى مثال، شاخص مان)، در حاليكه وظيفه طبقه بندى يك متغير هدف اسمى را استفاده مى كند كه با گروه محدودى از T ارزش هاى ممكن است. روش ها و تكنيك هاى متفاوت براى اين دو مدل از مسائل استفاده مى شوند. زيربناي بايد بردارى يا مترىك باشد، مثلاً xts به مسير اطلاعات عددى هدايت مى شود. اسلات اطلاعات شىء xts پكيچ روش اطلاعاتى واحد. اين بدین معنى است كه اين ممكن نيست كه يك ستون اطلاعات آموزشيمان را همچون (ما به اين سختى با اجرا R يك متغير عددى با تمام پيش بينى كننده هاى عددى داشته باشيم (يك فاكتر در غلبه خواهيم كرد. همچنان كه مشاهده خواهيم كرد xts كردن تمام تمامى مراحل مدل سازى خارج از چارچوب

معمولا برای اطلاعات تنظیمات فرعی و xts این کار آسان است و محدودیتی ندارد. زیربنای فراهم شده بوسیله ترسیمی استفاده می‌شوند اما مرحله مدل‌سازی برای این تسهیلات مورد نیاز نیست. کد زیر تمام دستورالعمل‌های اطلاعاتی که ما برای این بخش‌های متعاقب جهت بدست آوردن مدل‌های پیش‌بینی شده برای دو وظیفه استفاده می‌کنیم را ایجاد می‌کند.

```
> Tdata.train <- as.data.frame(modelData(data.model,
+ data.window=c('1970-01-02','1999-12-31'))))
> Tdata.eval <- na.omit(as.data.frame(modelData(data.model,
+ data.window=c('2000-01-01','2009-09-15'))))
> Tform <- as.formula('T.ind.GSPC ~.')
```

دو چارچوب اطلاعاتی با اطلاعات بدست آمده برای آزمایش و دوره‌های Tdata.train و Tdata.eval ارزیابی مربوطه هستند. ما چارچوب اطلاعات را همچون ساختار اطلاعات پایه‌ای برای بدست آوردن مدهای ترکیبی اطلاعاتی که در وظیفه طبقه‌بندی مورد نیاز خواهد بود، اجازه داده می‌شود. برای این وظایف ما ستون تولید خواهد شد، جایگزین trading.signals() ارزش هدف را با سیگنال‌های متناظری که با استفاده از تابع از تمام مدل‌های انتخاب شده و سنجش فرایندهایی که ما اجرا Tdata.eval می‌کنیم. چارچوب اطلاعات می‌کنیم، حذف خواهد شد. این در ارزیابی نهایی بهترین مدل‌هایی که ما انتخاب کردیم استفاده خواهد شد. در پایان چارچوب اطلاعاتی استفاده شده بوسیله فقدان اطلاعات NAs برای اجتناب از na.omit() فراخوان ضروری است. T آتی برای محاسبه شاخص

3-3-4- معیارهای ارزیابی

وظایف ارزیابی شرح داده شده در بخش‌های قبلی می‌تواند برای بدست آمدن مدل‌هایی که بعضی از نشانه‌ها در ارتباط با مسیر بازار را خروجی خواهد داد، استفاده شود. این علامت یک عدد در این مورد از وظایف پیش‌بینی شده)، یا نشانه مستقیمی در ارتباط با طبقه‌بندی وظایف است. حتی T رگرسیونی خواهد بود (ارزش در ارتباط با وظایف رگرسیون مشاهده کردیم که ما این عدد را بوسیله یک مکانیزم آستانه درون یک سیگنال مطرح شد. در بخش 3-5 چندین استراتژی تجاری که این علامت‌های پیش‌بینی را برای اجرا در یک بازار استفاده می‌کند را شرح خواهیم داد. در این بخش به سوالی از اینکه چگونه علامت‌های پیش‌بینی مدل‌هایمان را از پیش‌بینی‌های ارزیابی عددی را بررسی نخواهیم کرد. T ارزیابی کنیم آدرس‌دهی خواهیم کرد. ما شاخص طبق راهی که این پیش‌بینی‌های عددی را استفاده می‌کنیم، این ارزیابی کمی نامربوط است. یک نفر ممکنه سوال کند که آیا این منطقی است که این وظایف بازده را در حالیکه آن‌ها را تنها به سیگنال‌های مبادلات علاقه دارد، داشته باشیم. ما تصمیم می‌گیریم که این وظایف عددی بع علت استراتژی‌های متفاوت تجاری که می‌تواند مزیتی پیش‌بینی‌های عدد محسوب شود باقی بماند، به عنوان مثال، برای تصمیم‌گیری اینکه کدام مقدار پولی بسیار بیشتر از T برای سرمایه‌گذاری در زمانی که یک پوزیشن باز می‌شود مناسب است. برای مثال، ارزش

برای فروش) می‌تواند به سرمایه‌گذاری قویتری هدایت‌مان $T < -0.1$ برای خرید و $T > 0.1$ آستانه ما در عملکرد (کند. همچنانکه در زیر تعریف می‌شود، ارزیابی این علامت‌های پیش‌بینی می‌تواند بوسیله اندازه‌گیری نرخ خطا می‌تواند اجرا گردد:

$$error.rate = \frac{1}{N} \sum_{i=1}^N L_{0/1}(y_i, \hat{y}_i)$$

$L_{0/1}$ دارد و y_i است، که مقدار صحیح از کلاس برچسب i پیش‌بینی مدل برای آزمون مورد y_i در اینجا شناخته شده است: $0/1$ همچون تابع ضرر

$$L_{0/1}(y_i, \hat{y}_i) = \begin{cases} 1 & \text{if } \hat{y}_i \neq y_i \\ 0 & \text{if } \hat{y}_i = y_i \end{cases}$$

1- یک نفر اغلب از تکمیل کردن این اندازه‌گیری با عنوان شاخه شده صحت و دقت داده شده بوسیله استفاده می‌کند. $error.rate$

روز آتی اتفاق می‌افتد مقایسه k این دو آمار بطور اساسی با مدل پیش‌بینی که چه چیزی واقعا در بازار در می‌شود.

مشکل با صحت (یا نرخ خطا) این است که آن را حذف می‌کند نه اینکه برای این مدل از مشکلات به عنوان یک اندازه‌گیری خوب باشد. در نتیجه، اینجا بین سه خروجی محتمل با یک درجه نفوذ قوی از علامت‌های نگه داشتن بیش از دوتای دیگر عدم توازن بسیار قوی مانند حرکت‌های بزرگی در قیمت‌ها که پدیده نادری در بازارهای مالی است، وجود خواهد داشت. این بدین معنی است که رتبه دقت و صحت بوسیله عملکرد مدل‌های روی بیشترین تکرار خروجی که نگه داشتن است تسلط خواهد داشت. این امر در تجارت چندان خوشایند نیست. ما می‌خواهیم مدل‌هایی داشته باشیم که در علامت‌های نادر دقیق هستند (خرید و فروش). اینها چیزهایی هستند که فعالیت‌های بازار را هدایت می‌کند و بنابراین سود بالقوه هدف نهایی این برنامه است. پیش‌بینی‌های بازار مالی مثالی از یک برنامه گردنده بوسیله اتفاقات نادر است. وظایف پیش‌بینی بر مبنای وقایع معمولا بوسیله دقت و استانداردهای متریک فراخوانی شده که متمرکز بر ارزیابی اتفاقات، عدم رعایت عملکرد شرایط عادی است، ارزیابی می‌شوند (در نمونه ما، علامت‌های نگه داشتن). دقت به طور رسمی می‌تواند همچون درجه‌ای از سیگنال اتفاقات تولید شده بوسیله مدل‌هایی که صحیح هستند تعریف شود. فراخوان نیز درجه اتفاق افتادن رویدادها در قلمرویی که سیگنال‌ها به مانند همین مدل است تعریف می‌شود. این استانداردهای متریک می‌تواند به سادگی با کمک اغتشاش متریک که نتایج یک مدل را در اصطلاحی از مقایسه بین پیش‌بینی‌ها و ارزش‌های واقعی برای دستگاه آزمون ویژه محاسبه می‌کند، اندازه‌گیری شود. جدول 3-1 نشان داده شده یک مثال از اغتشاش ماتریک برای قلمروی مان است.

جدول 3-1: یک ماتریکس درهم برهم برای پیش‌بینی سیگنال‌های مبادلاتی

		Predictions			
		sell	hold	buy	
True Values	sell	$n_{s,s}$	$n_{s,h}$	$n_{s,b}$	$N_{s,}$
	hold	$n_{h,s}$	$n_{h,h}$	$n_{h,b}$	$N_{h,}$
	buy	$n_{b,s}$	$n_{b,h}$	$n_{b,b}$	$N_{b,}$
		$N_{.,s}$	$N_{.,h}$	$N_{.,b}$	N

با کمک جدول 1-3 ما می‌توانیم مفهوم دقت و فراخوانی را برای این مسئله مانند زیر شکل دهیم:

$$Prec = \frac{n_{s,s} + n_{b,b}}{N_{.,s} + N_{.,b}}$$

$$Rec = \frac{n_{s,s} + n_{b,b}}{N_{s,} + N_{b,}}$$

ما همچنین می‌توانیم این آمار را برای نشانه‌هایی ویژه بوسیله بدست آوردن دقت و فراخوانی برای علامت‌های خرید و فروش، به طور مستقل ارزیابی کنیم؛ برای مثال،

$$Prec_b = \frac{n_{b,b}}{N_{.,b}}$$

$$Rec_b = \frac{n_{b,b}}{N_{b,}}$$

نامیده می‌شود F-measure دقت و فراخوانی اغلب با یک سیگنال آماری ادغام می‌شود که ، که بدین صورت است: (Rijsbergen, 1979)

$$F = \frac{(\beta^2 + 1) \cdot Prec \cdot Rec}{\beta^2 \cdot Prec + Rec}$$

، ارتباط اهمیت فراخوانی و دقت را کنترل می‌کند. $0 \leq \beta \leq 1$ که در اینجا

3-4_مدل‌های پیش‌بینی

در این بخش ما برخی مدل‌هایی که می‌تواند برای آدرس‌دهی وظیفه پیش‌بینی تعریف شده در بخش قبلی استفاده شده را کاوش خواهیم کرد. این انتخاب مدل‌ها اساساً بوسیله این حقیقت که این تکنیک‌ها به خوبی توسط توانایی‌شان برای مدیریت کردن مسائل رگرسیون غیرخطی عالی‌تر شناخته می‌شوند، راهنمایی می‌کند. که این مورد مشکل مسئله ما می‌باشد. هنوز روش‌های بسیار می‌تواند در مورد این مسئله اجرا شود. هر دیدگاه کاملی در این قلمرو الزاماً یک مقیاس بزرگتری از تناوب بیشتر است. در متن این کتاب چنین کاوشی با توجه به هزینه آن در دوره زمانی و محاسبات قوی مورد نیاز منطقی نیست.

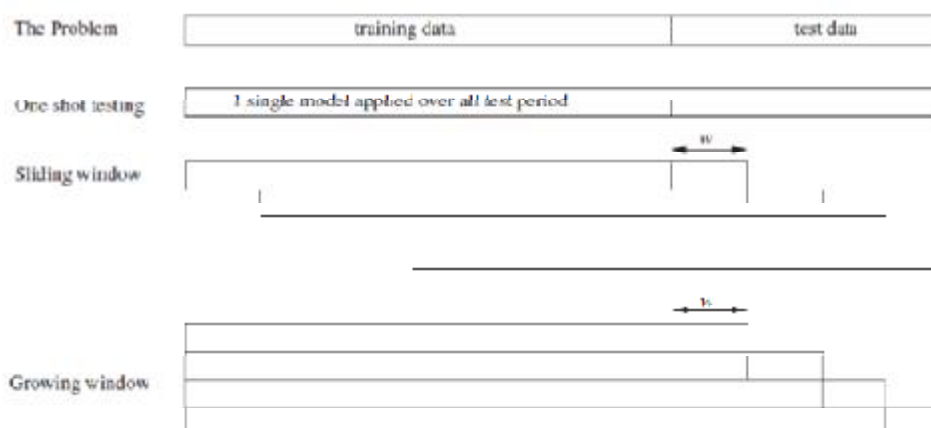
3-4_1- چگونه رشته اطلاعات استفاده می‌شوند؟

مشکلات سری‌های زمانی پیچیده به طور مکرر رژیم‌های متفاوتی مانند دوره‌هایی با تغییرات قوی پیامدی در دوره‌های با ثبات‌تر یا دوره‌هایی با بعضی شکل‌های گرایش سیستماتیک را ارائه می‌دهند. این شکل از

پدیده‌ها اغلب متحرک (پویا) نامیده می‌شوند و مسائل جدیدی را برای تکنیک‌های چندین مدل بر طبق فرضیات اصولیشان، ایجاد می‌کنند. از نظر منطقی این در مشاهده آسان است، مثلاً با ترسیم سری‌های زمانی قیمت که موردی برای اطلاعات ما است. اینجا چندین استراتژی وجود دارد که می‌توانیم دنبال کنیم تا سعی کنیم بر اثرات فشار منفی غلبه کنیم. مثلاً، چندین تکنیک انتقال می‌تواند نسبت به سری‌های زمانی اصلی برای حذف کردن بعضی اثرات اجرا شود. استفاده از درصد تغییرات (بازده) در عوض ارزش قیمت خالص اصلی یک نمونه از این مثال است. روش‌های دیگر شامل استفاده از دیتای در دسترس یک راه برگزیده‌تر است. اجازه دهید فرض کنیم به ما وظیفه بدست آوردن یک مدل با استفاده از دوره معینی از اطلاعات آموزشی داده شده است و سپس آن را در یک دوره متعاقب تست می‌کنیم. استاندارد دستیابی در اطلاعات آموزشی برای بسط دادن مدلی که بتواند برای بدست آوردن پیشگویی‌هایی برای دوره آزمایشی بکار رود، استفاده خواهد شد. چنانچه دلایل قوی برای باور کردن اینکه تغییرات رژیم وجود دارد وجود داشته باشد با استفاده از همان مدل در تمام دوره‌های آزمایشی ممکن است بهترین نظریه نباشد خصوصاً اینکه در خلال این دوره رژیمی تغییر کند آن می‌تواند به طور جدی عملکرد مدل را خراب کند. در این موارد اغلب بهتر است مدل را با استفاده از درصد اطلاعاتی که اطلاعات رژیم جاری را بهتر تسخیر می‌کند تغییر دهیم یا سازگار کنیم. در مشکلات سری‌های زمانی یک سفارش ضمنی (زمان) میان موارد تست شده وجود دارد. در این متن، این منطقی است که فرض کنیم که وقتی قبلاً به زمان $k < i$ بدست می‌آوریم، همه موارد آزمایش شده با برجسب زمان آداریم یک پیش‌بینی برای زمان گذشته تعلق داشته است. این یعنی که این امن است که فرض کنیم که ما قبلاً ارزش حقیقی متغیر هدف این موارد تست شده در گذشته را می‌دانستیم و بعلاوه می‌توانیم با امنیت از این اطلاعات استفاده کنیم. بنابراین، اگر از دوره تست ما مطمئنیم که آنجا رژیم به سری‌های زمانی تغییر مکان می‌یابد سپس ما می‌توانیم m در زمان را درون رشته اطلاعات اولیه متحد کنیم و با این تازه کردن m اطلاعات تمام موارد تست اتفاق افتاده قبلی گروه آموزشی که شامل مشاهدات رژیم جدید است تا حدی مدل‌های پیش‌بینی‌مان را برای توسعه عملکرد موارد آزمایشی آتی به روز رسانی کنیم. یک شکل از به روز رسانی مدل می‌تواند به تغییر آن در سفارشی که به حساب موارد جدید آموزشی قرار می‌گیرد، باشد. این روش‌ها معمولاً به عنوان یک عامل رشد کارآموزانی که با مدل اخیر به عنوان شاهد جدید وفق یافته‌اند به جای آنکه از صفر شروع کنند شناخته می‌شود. اینجا ، استفاده کرد. در این متن ما R تکنیک‌های مدل‌سازی فراوانی وجود ندارد که بتوان در این شیوه، خصوصاً در سایر روش‌ها برای به روز کردن مشکل را دنبال می‌کنیم که مشتمل بر یادگیری مجدد یک مدل جدید با استفاده از بسته آموزشی به روز شده می‌باشد. این روش به طور آشکاری در اصطلاح رقابتی گران‌تر است و ممکن است حتی برای نرم‌افزارهایی که اطلاعات را خیلی سریع دریافت می‌کنند و هر کدام از مدل‌ها و تصمیماتی که بلادرنگ مورد نیاز است، نامناسب باشد. این مسئله که تا اندازه‌ای در آدرس‌دهی برنامه‌ها در یک منطقه جستجو تکرار می‌شود عموماً همچون جریان اطلاعات شناخته می‌شود. در نرم‌افزار ما، بر مبنای روزانه پس از آنکه بازار بسته شد تصمیم‌گیری می‌کنیم بنابراین سرعت اهمیت چندانی ندارد. با فرض این که ما یک هدف یادگیری مجدد را استفاده خواهیم کرد، دو فرم اساسی از جا دادن موارد جدید به بسته آموزشی‌مان داریم.

پنجره رشد هدف به سادگی آن‌ها را به بسته آموزشی جدید اضافه می‌کند، بنابراین با افزایش ثابت اندازه این بسته مواجه می‌شویم. مشکل احتمالی این هدف بر این حقیقت که فرض می‌کنیم که بیشتر درصد اطلاعات لازم است در تولید مدل‌های بهتر استفاده شود، تکیه دارد همچنین ما ممکن است بررسی کنیم که بخش‌های قدیمی اطلاعات آموزشی مان ممکنه خیلی قدیمی باشد و در نتیجه دقت مدل‌ها کاهش یابد. بر اساس این بررسی پنجره اهداف شامل یک تصمیم کلیدی است: وقتی که مدل بوسیله به هم پیوستن اطلاعات تازه‌تر تغییر می‌کند یا سازگار می‌شود. آنجا ضرورتاً دو روش پاسخ‌دهی به این سوال وجود دارد. اولین شامل تخمین زدن این زمان بوسیله چک کردن اینکه عملکرد مدل جاری مان دارد نزول می‌کند، می‌شود. اگر ما یک کاهش ناگهانی در این عملکرد مشاهده کنیم سپس می‌توانیم این را به عنوان یک نشان خوبی از بعضی رژیم تغییر یافته در نظر بگیریم. مبارزه اصلی در این دستیابی بر توسعه مناسب تخمینی این تغییرات در عملکرد تکیه دارد. ما می‌خواهیم تغییرات را هر چه زودتر کشف کنیم اما نمی‌خواهیم واکنش اغراق‌آمیزی به برخی تست‌های تقلبی مسئله که مدل مان از دست داده داشته باشیم. هدف ساده‌تر دیگر شامل به روز کردن مدل بر مبنای زمان منظم می‌باشد، ما یک مدل جدید با اطلاعات تازه‌تر بدست می‌آوریم. در این مورد آموزشی W است که هر نمونه تست ما باشد، ما این روش ساده را دنبال می‌کنیم.

به طور خلاصه، برای هر مدلی که بررسی می‌کنیم، ما آن را با استفاده از سه روش متفاوت اجرا خواهیم کرد: 1) مدل تکی برای تمام دوره‌های آموزشی، 2) پنجره رشد با یک مرحله ثابت به روز رسانی از روزهای 3-3. نمودار این سه روش را توضیح می‌دهد. 3) پنجره کشویی با همان مرحله به روز رسانی



نمودار 3-3: سه شکل از پیش‌بینی‌های بدست آمده برای یک دوره آزمایشی

مطالعه بیشتر در مورد تغییرات رژیم

مشکل کشف کردن تغییرات از اطلاعات سری‌های زمانی رژیم یک موضوع مطالعه برای زمانی طولانی در ، که از تکنیک‌هایی مانند کنترل (Oakland, 2007) محیطی شناخته شده به عنوان کنترل فرایند آماری بود

چارت‌ها برای شناسایی نقاط شکسته شده اطلاعات استفاده می‌کرد. این موضوع بر یک افزایش علاقه‌مندی به در رشته داده‌کاوی گواهی می‌دهد. چندین کار (برای (Gama and Gaber, 2007) فشار جریان اطلاعات (نظریه چگونگی کشف تغییرات (Gama et al., 2004; Kifer et al., 2004; Klinkenberg, 2004 مثال رژیم و همچنین چگونگی یادگیری مدل‌های یادگیری در حضور این تغییرات را نشان می‌دهد.

3-4-2 ابزارهای مدل‌سازی

در این بخش به طور خلاصه تکنیک‌های مدل‌سازی را که در آدرس‌دهی وظایف پیش‌بینی‌مان استفاده شرح می‌دهیم. R می‌کنیم و چگونگی استفاده از آن‌ها را در

3-4-2-1 شبکه‌های عصبی مصنوعی¹

شبکه‌های عصبی مصنوعی به دلیل قابلیت‌شان برای رسیدگی کردن به مسائل غیر خطی زیادی مکرراً در را اجرا می‌کند. این شکل R پیش‌خواند شبکه‌های عصبی در nnet پیش‌بینی‌های مالی استفاده می‌شوند. پکیج از شبکه‌های عصبی در بین بیشترین استفاده‌ها قرار دارند و نیز جزو آندسته‌ای هستند که برای ما بیشترین کاربرد را دارد.

شبکه‌های عصبی مصنوعی بوسیله یک گروه از واحدهای محاسبه (نرون‌های عصبی) به همدیگر متصل هستند. هر نرون دو دو محاسبه پیاپی را اجرا می‌کند: یک ترکیب خطی از ورودی‌ها که با یک ترکیب غیرخطی از نتایج بدست آمده از ارزش خروجی‌ها دنبال می‌شود که سپس باقی نرون‌های شبکه را تغذیه می‌کند. هر کدام از اتصالات نرون یک وزن وابسته دارد. ساختار یک شبکه عصب مصنوعی عبارت است از ساختن یک معماری برای شبکه و سپس استفاده از یک الگوریتم برای پیدا کردن وزنه‌های ارتباط بین نرون‌ها می‌باشد. تغذیه رو به جلوی شبکه‌های عصبی مصنوعی نرون‌های عصبی را در لایه‌ها سازماندهی می‌کند. اولین لایه شامل نرون‌های ورودی به شبکه است. مشاهدات آموزشی این مسئله در شبکه از طریق این نرون‌های خروجی ارائه شده‌اند. لایه آخری شامل پیش‌بینی‌های شبکه عصبی برای هر مورد ارائه شده در این نرون‌های ورودی است. در این میان، ما معمولاً یک یا چند لایه عصبی مخفی داریم. وزن الگوریتم‌های به روز رسانی، مانند الگوی انتشار به عقب، سعی دارد برای بدست آوردن یک اتصال وزنی که یک معیار خطای مشخصی را بهینه‌سازی کند. این به وسیله یک فرایند تکراری از زمان‌های چندگانه ارائه موارد آموزشی در گره ورودی شبکه انجام می‌شود و بعد از بدست آمدن پیش‌بینی شبکه در گره‌های خروجی و محاسبه خطاهای پیش‌بینی مربوطه، معیارهای وزنی در شبکه به روز رسانی می‌شود تا سعی شود خطاهای پیش‌بینی بهبود یابد. این فرایند تکراری مرتب تکرار می‌شود تا به بعضی با استفاده از یک R معیارهای همگرایی دست یابیم. تغذیه رو به جلو با یک لایه مخفی می‌تواند به راحتی در شبکه‌های بدست آمده بوسیله این (Venables and Ripley, 2002) بدست آید nnet تابع از پکیج تابع می‌تواند با هر دو مدل طبقه‌بندی و مسائل رگرسیون بدست آید و بنابراین با هر دو وظیفه پیش‌بینی‌مان به گونه‌ای شناخته شده هستند که به مقیاس‌های ANN قابل دست‌یابی است (به بخش 3-3-3 توجه کنید).

¹ - شبکه‌های عصبی مصنوعی منظور از سیستم‌های کامپیوتری است که از برنامه‌ها و اطلاعات ساخته شده که شامل چندین پردازش موازی است و نوعی از شبیه‌سازی است که مانند ساختار مغز انسان کار می‌کند

متفاوتی از متغیرهای استفاده شده در یک برنامه پیش‌بینی حساس باشند. در این متن، این منطقی است که اطلاعات را قبل از دادن آن‌ها به شبکه به خاطر اجتناب از فشار منفی احتمالی در عملکرد، تبدیل کنیم. در نمونه‌مان، ما اطلاعات را با هدف ساختن تمام متغیرهایی که ارزش پایین‌تر از صفر دارند و انحراف معیار از آن استاندارد است، هنجار سازی می‌کنیم. این می‌تواند به آسانی بوسیله نقل و انتقالات زیرین که در هر ستون اطلاعات تنظیم شده بکار رود.

$$y_i = \frac{x_i - \bar{x}}{\sigma_x}$$

می‌تواند برای اجرا `scale()` انحراف استاندارد است. تابع σx اصلی و x کمترین ارزش متغیر \bar{x} جایی که را بیابید `unscale()` کردن این انتقالات برای اطلاعات مان استفاده شود. در پکیج کتاب همچنین می‌تواند تابع که فرایند نرمال سازی گذاشته شده در ارزش‌های قبلی مقیاس اصلی را معکوس می‌کند. در زیر شما می‌توانید را مشاهده نمایید: R در ANN یک توضیح بسیار ساده از چگونگی بدست آوردن و استفاده از این نوع

```
> set.seed(1234)
> library(nnet)
> norm.data <- scale(Tdata.train)
> nn <- nnet(Tform, norm.data[1:1000, ], size = 10, decay = 0.01,
+   maxit = 1000, linout = T, trace = F)
> norm.preds <- predict(nn, norm.data[1001:2000, ])
> preds <- unscale(norm.preds, norm.data)
```

وزن‌های اولیه از لینک‌های بین گره با ارزش‌های تصادفی در فاصله `[0/5]` `nnet()` به طور پیش‌فرض تابع `.... -0/5` را تنظیم می‌کند. این یعنی که دو حرکت متوالی از تابع با استدلال کاملاً شبیه می‌تواند به نتایج حقیقی متفاوتی رهنمون سازد. آنچنان که در ادامه مشاهده می‌کنید، برای اینکه شما را از بدست آمدن نتایج اضافه کنیم که به مولد عدد تصادفی برای بعضی `set.seed()` مشابه مطمئن سازیم باید یک فراخوان به تابع مانند آنچه اینجا ANN مقادیر مشاهده شده مقدار اولیه می‌دهد. این اطمینان می‌دهد که شما دقیقاً همان گزارش کرده‌ایم را بدست می‌آورید. در این مثال واضح ما باید از اولین 1000 مورد برای بدست آوردن شبکه و را `nnet()` تست مدل در روی 1000 تای بعدی استفاده کنیم. بعد از نرمال سازی اطلاعات آموزشی مان، ما تابع می‌باشد: فرم‌های R برای بدست آوردن مدل فراخوانی می‌کنیم. اولین دو پارامتر معمولاً هر تابع مدل سازی در تابعی از مدل بوسیله یک فرمول مشخص می‌شود و مثال‌های آموزشی برای بدست آوردن مدل استفاده می‌شود. استفاده می‌کنیم. به طور مثال اندازه پارامتر به ما اجازه می‌دهد `nnet()` ما همچنین از بعضی پارامترهای تابع مشخص کنیم چه تعداد گره در لایه‌های مخفی وجود خواهد داشت. اینجا هیچ دستورالعمل جادویی وجود ندارد

که کدام ارزش را اینجا استفاده کنیم. یک نفر معمولاً برای مشاهده رفتار شبکه چندین ارزی (بها) را بررسی می‌کند. هنوز این منطقی است که فرض کنیم این باید کوچکتر از تعداد پیش‌بینی کننده‌های مسئله باشد. پارامترهای تنزل، نرخ به روز رسانی وزنی را از الگوریتم انتشار قبلی کنترل می‌کند. دوباره در اینجا نیز، آزمون و تعداد بیشینه تکرار فرایند همگرایی وزنی \maxit خطا بهترین دوست ما است. در نهایت کنترل‌های پارامتر می‌گوید تابعی که ما مدیریت می‌کنیم یک مشکل $linout=T$ برای استفاده مجاز است، در حالیکه تنظیمات برای اجتناب از بعضی خروجی‌هایی که به بهینه‌سازی فرایند مربوط است، استفاده $Trace=F$ رگرسیونی دارد. می‌تواند برای بدست آوردن پیش‌بینی‌هایی از شبکه عصبی برای تنظیم اطلاعات $predict()$ می‌شود. تابع آزمایش شده، استفاده شود. پس از بدست آوردن این پیش‌بینی‌ها ما آن‌ها را به حالت مقیاس اصلی استفاده که بوسیله پکیج ما فراهم شده است، تبدیل می‌کنیم. این تابع در اولین استدلال $unscale()$ شده در تابع ارزش‌ها دریافت می‌شود و در دومین استدلال هدف با اطلاعات نرمال سازی می‌شود. این هدف بعدی ضروری است زیرا درون آن هدف است که انحرافات استاندارد و میانگین که ما برای نرمال سازی اطلاعات استفاده کرده بودیم، ذخیره شد و اینها برای دستیابی به نرمال سازی معکوس مورد نیاز است.

برای پیش‌بینی صحت علامت‌ها برای گروه آزمایشی را ارزیابی کنیم. ما این کار را ANN اجازه دهید نتایج بوسیله تبدیل کردن پیش‌بینی‌های اعداد به علامت‌ها و سپس ارزیابی آن‌ها با استفاده از آمار ارائه شده در بخش 3-4 انجام می‌دهیم.

```
> sigs.nn <- trading.signals(preds, 0.1, -0.1)
> true.sigs <- trading.signals(Tdata.train[1001:2000, "T.ind.GSPC"],
+ 0.1, -0.1)
> sigs.PR(sigs.nn, true.sigs)
```

	precision	recall
s	0.2101911	0.1885714
b	0.2919255	0.5911950
s+b	0.2651357	0.3802395

پیش‌بینی‌های عددی را با دادن آستانه خرید و فروش مربوطه به علامت‌ها $trading.signals()$ تابع یک ماتریکس با امتیاز فراخوانی و دقت از این دو نوع وقایع و کلیات بدست $sigs.PR()$ تبدیل می‌کند. تابع درخشان نیست. در نتیجه، شما امتیازات با دقت کمتری ANN می‌آورد. این امتیازات نشان می‌دهد که عملکرد بدست می‌آورد و همچنین خیلی به ارزش‌های فراخوانی علاقه‌مند نیست. از آنجا که آن‌ها اساساً به معنی از دست دادن فرصت‌ها است نه قیمت‌ها بنابراین این یک مسئله جدی نیست. برعکس امتیازهای دقت پایین بدین معنی است که مدل علامت‌های اشتباه نسبتاً مکرری می‌دهد. اگر این علامت‌ها برای مبادلات استفاده شود همچنین می‌تواند برای وظایف طبقه‌بندی استفاده ANN ممکنه به از دست دادن‌های جدی مالی هدایت شویم.

شود. برای این مشکلات، اختلاف اصلی در اصطلاحات شبکه تکنولوژی این است که به جای یک واحد خروجی تک، ما واحدهای خروجی بیشماری مانند آنچه که ارزش متغیر هدف است خواهیم داشت (گاهی اوقات به عنوان کلاس متغیر شناخته می‌شود). هر کدام از این واحدهای خروجی یک احتمال تخمینی از ارزش کلاس می‌تواند به عنوان ANN مربوطه را تولید خواهد کرد. این بدین معنی است که برای هر کدام از این موارد یک یک گروه احتمالی ارزش تولید شود، یکی برای هر کلاس ارزش ممکن. برای این وظایف بسیار شبیه استفاده آن برای مشکلات رگرسیون است. کد زیر این `nnnet()` استفاده از تابع مسئله با استفاده از اطلاعات آموزشی ما توضیح می‌دهد:

```
> set.seed(1234)
> library(nnet)
> signals <- trading.signals(Tdata.train[, "T.ind.GSPC"], 0.1,
+   -0.1)
> norm.data <- data.frame(signals = signals, scale(Tdata.train[,
+   -1]))
> nn <- nnet(signals ~ ., norm.data[1:1000, ], size = 10, decay = 0.01,
+   maxit = 1000, trace = F)
> preds <- predict(nn, norm.data[1001:2000, ], type = "class")
```

برای بدست آوردن یک برچسب کلاس واحد برای هر کدام از موارد آزمایش در `type="class"` آرگومان "کلاس" عوض گروه تخمینی احتمالی استفاده شده است. با پیش‌بینی‌های شبکه ما می‌توانیم دقت و فراخوانی مدل را همچون زیر ارزیابی کنیم:

```
> sigs.PR(preds, norm.data[1001:2000, 1])
```

	precision	recall
s	0.2838710	0.2514286
b	0.3333333	0.2264151
s+b	0.2775665	0.2185629

هر دو امتیاز دقت و فراخوانی بالاتر از آن است که در وظیفه رگرسیون بدست آمده اگر چه هنوز ارزش (بها) پایین است.

مطالعه بیشتر در مورد شبکه‌های عصبی

در سال 1996 یک منبع عمومی معتبری است که در مورد شبکه‌های Rojas کتاب نوشته شده توسط یک کتاب خوب Zirilli عصبی وجود دارد. برای مطالعات دقیق‌تر مالی، کتاب نوشته شده در سال 1997 توسط و آسان است. مجموعه صفحات ملقب به "سری‌های زمانی پیش‌بینی شبکه‌های عصبی مصنوعی" نوشته راجرز و

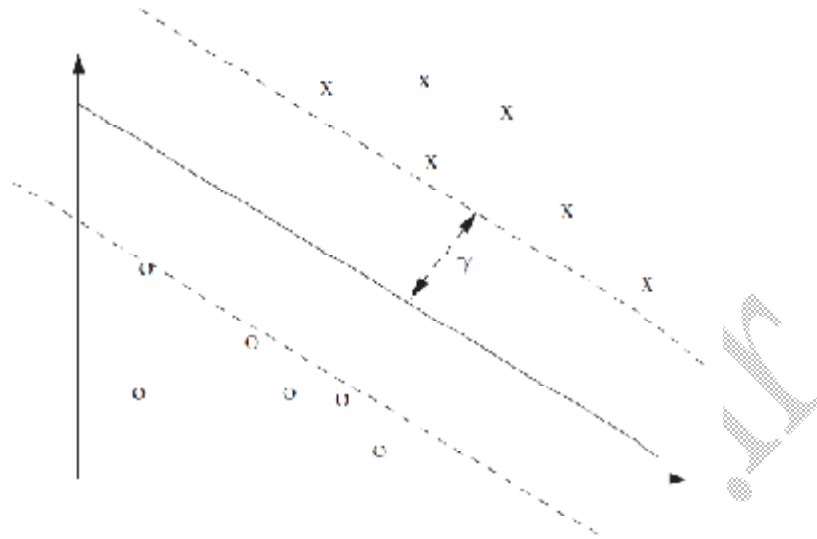
وموری در سال 1994 نیز مثال دیگر از منابع اطلاعاتی مفید است. بخش 1 از کتاب نوشته شده توسط دبوک در سال 1994 چندین فصل فراهم می‌کند که به برنامه‌های مبادلاتی در شبکه‌های عصبی اختصاص داده شده است. کار مک‌کلاچ و پیتز در سال 1943 اولین مدل شبکه‌های مصنوعی را ارائه می‌کند. این کار بوسیله رونزنبلات در سال 1958 و مینسکی و پاپرت در سال 1969 تعمیم یافته است. الگوریتم انتشار به قبل، بیشترین استفاده را در روش به روز رسانی وزنی دارد اگرچه خیلی وقت‌ها به راملهارت در سال 1986 منسوب است که بر طبق نظریات راجز در سال 1996 و اختراع شده توسط ورز در سال 1974 تا 1996 می‌باشد.

3-2-4-2 مکانیسم بردار حمایت

می‌تواند به هر دو وظیفه ANNs ابزار مدل‌سازی است که مانند (SMVs) مکانیسم بردار حمایت گواهی توجهات افزایش یافته از انجمن‌های تحقیقاتی مختلف بر SVMs طبقه‌بندی و رگرسیون بکار برده شود. پایه موفقیت نرم‌افزارشان برای قلمروهای مختلف و نیز سوابق تئوری قوی‌شان می‌باشد. واپنیک (در سال 1995 هستند. اسمولا و اسکولکوف SVMs تا 1998) و شاو-تایلر و کریستیانی (سال 2000) دو منبع ضروری برای SVMs (سال 1998 تا 2004) یک خودآموز بسیار عالی با دادن یک مرور مختصری از نظریات اصولی ابتدایی در دسترس داریم، که از جمله آن‌ها می‌توانیم به پکیج SMVs ما چندین ابزار R برای رگرسیون منتشر کرد. در e1071 در پکیج svm() توسط کاراتزگلو در سال 2004 با چندین تابع در دسترس و همچنین تابع kernlab بوسیله دیمیتزیدو (سال 2009) اشاره کنیم.

نقشه‌کشی اطلاعات اصلی درون یک فضای تازه چند بعدی است، که ممکن SVMs مقصود اولیه پشت استراتژی برای مدل‌های خطی جهت بدست آوردن یک برنامه افراطی جدا کننده بکار رود، برای مثال، مشکل جداسازی کلاس‌ها در وظیفه طبقه‌بندی موارد. نقشه‌برداری اطلاعات اصلی این فضای جدید با کمک کدایی تابع ماشین‌های خطی اجرایی بر روی این استنتاج دوگانه ارائه شده بوسیله توابع کرنل SMVs کرنل اجرا می‌شود. است.

جدا کننده خطی بالاتر در ارائه دوگانه جدید معمولاً بوسیله افزایش حاشیه جدایی بین موارد متعلق به کلاس‌های مختلف می‌باشد؛ نمودار 3-4 را مشاهده کنید. این یک مشکل بهینه سازی شده است که اغلب بوسیله مدل‌های برنامه‌ریزی غیرخطی حل می‌شود. روش‌های حاشیه نرم به نسبت‌های کوچکتری از مورد اجازه می‌دهد که در سمت غلط حاشیه قرار بگیرد، که هر کدام از این‌ها به یک هزینه قطعی هدایت می‌کند.



SVMs نمودار 4-3: بیشینه‌سازی حاشیه در

در حمایت از رگرسیون برداری، فرایند شبیه است با اختلاف اساسی که بر روی شکل خطاها آغاز می‌شود و هزینه‌های وابسته محاسبه می‌گردند. این دسته‌بندی مجدد معمولاً برای استفاده از تابع زیان کذایی داده شده بوسیله فرمول زیر است: ϵ insensitive

$$|\xi|_{\epsilon} = \begin{cases} 0 & \text{if } |\xi| \leq \epsilon \\ |\xi| - \epsilon & \text{otherwise} \end{cases}$$

بیان خواهیم کرد. با وظیفه رگرسیون R ما هم‌اکنون یک مثال بسیار ساده از استفاده اینگونه مدل‌ها در که استفاده خواهد شد، آغاز خواهیم کرد: e1071 برای هر کدام از توابع فراهم شده در پکیج

```
> library(e1071)
> sv <- svm(Tform, Tdata.train[1:1000, ], gamma = 0.001, cost = 100)
> s.preds <- predict(sv, Tdata.train[1001:2000, ])
> sigs.svm <- trading.signals(s.preds, 0.1, -0.1)
> true.sigs <- trading.signals(Tdata.train[1001:2000, "T.ind.GSPC"],
+   0.1, -0.1)
> sigs.PR(sigs.svm, true.sigs)
```

	precision	recall
s	0.4285714	0.03428571
b	0.3333333	0.01257862
s+b	0.4000000	0.02395210

با بیشترین پارامترهای پیش فرض آن با استثنائات هزینه و محدوده $\text{svm}()$ در این مثال ما از تابع پارامترها استفاده می‌کنیم. در این متن تابع استفاده شده یک تابع کرنل رادیال مبنایی است.

$$K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma \times \|\mathbf{x} - \mathbf{y}\|^2)$$

پارامتر استفاده کننده است که در فراخوان مان آنرا با مقدار 0/001 تنظیم کرده‌ایم (تابع γ در جایی که (اطلاعات) را استفاده می‌کند). $1/\text{ncol}()$ یک مقدار پیش فرض $\text{svm}()$

پارامتر هزینه بر هزینه‌هایی از تخلف از حاشیه اشاره دارد. شما ممکنه مایل باشید صفحه کمک تابع را برای یادگیری جزئیات بیشتر در این مورد و سایر پارامترها مطالعه نمایید.

ANN به یک رتبه بهتر بسیار قابل ملاحظه نسبت به واژه دقت SVM ما می‌توانیم مشاهده کنیم که مدل دست یابد، اگرچه با یک فراخوان بسیار کمتر باشد.

استفاده می‌کنیم: `kernlab` در آینده، وظیفه طبقه‌بندی را بررسی می‌کنیم، این بار از پکیج

```
> library(kernlab)
> data <- cbind(signals = signals, Tdata.train[, -1])
> ksv <- ksvm(signals ~ ., data[1:1000, ], C = 10)
```

laplace یا کرنل RBF برای (sigest) با استفاده از تخمین اتوماتیک سیگما

```
> ks.preds <- predict(ksv, data[1001:2000, ])
> sigs.PR(ks.preds, data[1001:2000, 1])
```

	precision	recall
s	0.1935484	0.2742857
b	0.2688172	0.1572327
s+b	0.2140762	0.2185629

برای ویژه کردن یک هزینه متفاوت از محدودیت `kernlab` از پکیج `ksvm()` را برای تابع `C` ما پارامتر انحرافات که به طور پیش فرض مقدار آن 1 است، استفاده می‌کنیم. صرف نظر از اینکه ما از ارزش پارامتر پیش فرض استفاده کنیم که برای طبقه‌بندی مرتبط، برای مثال، استفاده از مبنای کرنل رادیال. بار دیگر، بدست SVM به جذابیت SVM می‌تواند بدست آید. نتیجه این `ksvm()` جزئیات بیشتر در بسته کمکی تابع آمده با اطلاعات رگرسیون نیست. ما باید خاطر نشان کنیم که به هیچ عنوانی نمی‌خواهیم ادعا کنیم که اینها

بهترین منابعی هستند که می‌توانیم از این تکنیک‌ها به دست آوریم. اینها تنها مثال‌های ساده‌ای هستند که را توضیح می‌دهد. R چگونگی استفاده از این تکنیک‌های مدل‌سازی در

3-2-4-3_ نوار رگرسیون چند متغیره تطبیقی

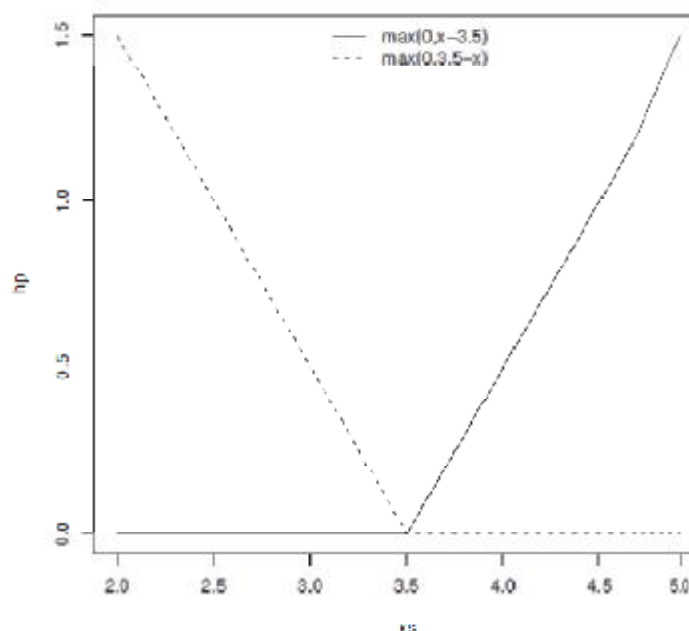
Hastie (2011) مثالی از یک مدل رگرسیون افزایشی (friedman, 1991) نوار رگرسیون چند متغیره تطبیقی یک فرم کلی است که در زیر می‌آید: MARS است. یک مدل (and Tibshirani, 1990)

$$mars(\mathbf{x}) = c_0 + \sum_{i=1}^k c_i B_i(\mathbf{x})$$

ها توابع پایه هستند. توابع پایه می‌توانند چندین شکل داشته باشند، از B_i ها اعداد ثابت و C_i در حالیکه اعداد ثابت ساده گرفته تا مدل‌های تابع متقابلی بین دو متغیر یا بیشتر. هنوز، توابع مبنایی متداول‌تری به عنوان توابع کذایی وابسته که شکل گرفته‌اند.

$$H[-(x_i - t)] = \max(0, t - x_i) \quad H[+(x_i - t)] = \max(0, x_i - t)$$

یک ارزش آستانه روی این پیش‌بینی است. نمودار 3-5 یک t یک پیش‌بینی کننده است و x_i جایی که مثال از این دو تابع را نشان می‌دهد.



نمودار 3-5: یک مثال از دو تابع وابسته با آستانه یکسان

شامل (Leisch et al., 2009) mda اجرا می‌شود. پکیج R در حداقل دو پکیج در MARS مدل‌های دارد که earth() تابع (Milborrow, 2009) است که در این روش اجرا می‌شود. پکیج زمین (mars) تابع در واژه توابع R همچنین در این تکنولوژی اجرا می‌شود. این تابع دوم مزیت پیرو استاندارد بیشتری از الگوی مدل‌سازی بوسیله فراهم کردن یک واسط بر مبنای فرمول را دارد. همچنین تسهیلات دیگر چندی را اجرا earth() می‌کند که در پکیج‌های دیگر ارائه نشده است و بنابراین این انتخاب ما خواهد بود. کدهای زیرین تابع را برای وظیفه رگرسیون اجرا می‌کند.

```
> library(earth)
> e <- earth(Tform, Tdata.train[1:1000, ])
> e.preds <- predict(e, Tdata.train[1001:2000, ])
> sigs.e <- trading.signals(e.preds, 0.1, -0.1)
> true.sigs <- trading.signals(Tdata.train[1001:2000, "T.ind.GSPC"],
+ 0.1, -0.1)
> sigs.PR(sigs.e, true.sigs)
```

	precision	recall
s	0.2785714	0.2228571
b	0.4029851	0.1698113
s+b	0.3188406	0.1976048

برای طبقه‌بندی بدست آمده با رتبه دقت حدود 30٪، هرچند با فراخوان پایین‌تری SVMs نتایج با آنکه با باشد، مقایسه می‌شود.

تنها برای مشکلات رگرسیونی قابل اجراست بنابراین ما هیچ مثالی برای وظیفه طبقه‌بندی نشان MARS نمی‌دهیم.

مطالعات آتی در نوار رگرسیون چند متغیره تطبیقی

مقاله نوشته شده توسط فریدمن در سال 1991 در مجله روزنامه است. این یک MARS منابع قطعی در را به خوبی تکنیک‌های MARS مقاله نگارش شده بسیار عالی است که تمام جزئیات مربوط به انگیزه‌ها توسعه استفاده شده در سیستم فراهم می‌کند. این مقاله همچنین شامل بحث کاملاً جالبی به وسیله دانشمندان دیگر است که چشم‌اندازهای دیگری از این کار را فراهم می‌کند.

3-5- از پیش‌بینی‌ها تا عملکردها

این بخش به نظریه اینکه چگونه علامت پیش‌بینی‌های بدست آمده بوسیله تکنیک‌های مدل سازی توصیف شده در قبل استفاده می‌شود، اختصاص دارد. یک گروه معین از علامت‌های خروجی بوسیله برخی مدل‌ها انجام می‌شود، اینجا روش‌های بسیاری وجود دارد که ما می‌توانیم از آن‌ها برای عمل کردن در یک بازار استفاده کنیم.

3-5-1- چگونه از پیش‌بینی‌ها استفاده خواهیم کرد؟

در مورد مطالعه ما، فرض خواهیم کرد در بازارهای آتی معامله خواهیم کرد. این بازارها بر مبنای قراردادهای خرید یا فروش یک کالا در یک تاریخ معین در آینده با قیمتی تعیین شده بوسیله بازار در زمان آینده است. جزئیات فنی این قراردادها فراتر از حوزه این دست‌نوشته (مطلب) است. هنوز، در دوره‌های هدف، این بدین معنی است که سیستم تجاری‌مان قادر خواهد بود دو نوع از موقعیت‌های تجاری بگشاید: بلند و کوتاه. موقعیت‌های $t+x$ باز می‌شود و فروش آن در زمان بعدی p و قیمت t طولانی مدتی بوسیله خریداران یک جنس در زمان است. این برای بازگشایی مبادله همچون موقعیت‌ها زمانی که او یک استثناء دارد که قیمت‌ها در آینده افزایش خواهد یافت، بنابراین اجازه می‌دهد در این مبادله مقداری سود کسب کند. در موقعیت‌های کوتاه مدت، بازرگان با تعهد خرید در آینده به طور محرمانه می‌فروشد. با تشکر از بروینگ اسکیمای کسی که p با قیمت t در زمان جزئیاتش را می‌توانید در مدارک فراهم شده در سایت ویکی‌پدیا بیابید این کار ممکن شد. این گونه از وضعیت‌ها خرید t اجازه می‌دهد بازرگان زمانی که قیمت‌ها کاهش می‌یابند نیز همچنانکه اون محرمانه در زمان بعدتر از می‌کند، سود کسب کند. به طور غیر رسمی، می‌توانیم بگوییم زمانی که باور داریم قیمت‌ها دارند افت می‌کنند موقعیت‌های کوتاه‌تری باز خواهیم کرد و موقعیت‌های طولانی‌تر را زمانی که باور داریم قیمت‌ها صعود می‌کنند، باز می‌کنیم.

با توجه به گروهی از علامت‌های داده شده، راه‌های بسیاری وجود دارد که می‌توانیم از آن‌ها در تجارت در بازارهای آتی استفاده کنیم. ما چند استراتژی تجاری پذیرفتنی و منطقی را شرح خواهیم داد که در تجربیاتمان با مدل‌ها استفاده و مقایسه می‌شوند. بر طبق محدودیت فضا و زمان، این ممکن نیست که این نظریه مهم را در آینده مورد جستجو و بررسی قرار دهیم. هنوز، خواننده با برخی استراتژی‌های پذیرفتنی و با مفهوم توسعه و دیگر احتمالات رها می‌شود.

مکانیسم استراتژی اولین تجارت که قصد داریم استفاده کنیم در ذیل بیان شده است. اول، تمام تصمیمات در پایان روز اخذ خواهند شد، که بعد از دانستن تمام قیمت‌های جاری روزانه از بخش جاری می‌باشد. فرض پیش‌بینی T ، مدل‌هایمان شواهدی فراهم می‌کنند که قیمت‌ها کم می‌شوند، که ارزش پایین t کنید در پایان روز می‌کنند یا یک علامت فروش. اگر هم‌اکنون تصمیم به باز کردن یک موقعیت داشته باشیم دلالت و نشانه مدل

نادیده گرفته خواهد شد. اگر اخیراً هیچ وضعیت بازگشایی نگه نداشته باشیم، ما یک موقعیت کوتاه بوسیله یک وقتی در آینده انجام pr نظریه یک دستور فروش باز خواهیم کرد. وقتی این سفارش بوسیله بازار در قیمت است، در $pr-p\%$ شد، فوراً دو دستور دیگر را پست می‌کنیم. اولی یک دستور خرید محدود با محدودیت قیمت یک هدف حاشیه سود می‌باشد. این نوع از دستورات تنها زمانی که قیمت بازار به محدوده قیمت $p\%$ جایی که هدف یا کمتر از آن دست می‌یابد قابل انجام است. این سفارش آن چیزی را که حاشیه سود هدف ما تنها برای موقعیت کوتاه مدت باز شده است را بیان می‌دارد. ما 10 روز برای رسیدن به این هدف صبر خواهیم کرد، در پایان دهمین روز ما در قیمت بسته شدن خرید خواهیم کرد. سفارش دوم یک دستور توقف خرید با محدوده است. این دستور با هدف محدود کردن زیان احتمالی با این موقعیت جایگزین می‌شود. اگر بازار $pr+l\%$ قیمت محدود می‌کنیم. اگر $l\%$ دست یابد دستور اجرا خواهد شد بنابراین زیان احتمالی ما را به $pr+l\%$ به قیمت یا نشانه‌های خرید، T مدل‌هایمان نشانه‌هایی فراهم کند که قیمت‌ها در آینده نزدیک با ارزش بالای پیش‌بینی افزایش خواهد یافت، یک موقعیت بلند مدت بررسی خواهیم کرد. این موقعیت تنها زمانی باز می‌شود که ما pr با قیمت it اخیراً خارج از قیمت بازار باشیم. با این هدف یک دستور خرید پست خواهیم کرد که در یک زمان به انجام خواهد رسید. همچون قبل، ما فوراً دو دستور جدید پست خواهیم کرد. اولی یک دستور محدود فروش یا بالاتر از آن $pr+p\%$ خواهد بود که تنها در حالتی اجرا می‌شود که بازار به قیمت $pr+p\%$ با یک قیمت هدف دست یابد. این دستور فروش محدود یک ضرب‌العجل 10 روزه، همچنانکه در قبل اشاره شد، خواهد داشت. محدود می‌کند. $l\%$ ، که دوباره زیانمان را به $pr-l\%$ دستور دومی یک دستور توقف فروش با قیمت

اولین استراتژی کمی محافظه‌کارانه دیده می‌شود همچنانکه آن تنها یک موقعیت بازگشایی در هر زمان دارد. بعلاوه، بعد از 10 روی انتظار برای سود هدف، موقعیت فوراً بسته می‌شود. ما همچنین یک استراتژی تجاری با ریسک بیشتر را بررسی می‌کنیم. این استراتژی دیگر شبیه قبلی است، با این استثنا که اگر نشانه‌هایی با آن علامت وجود داشته باشد و اگر پول کافی برای آن داشته باشیم، ما همیشه یک موقعیت جدید باز خواهیم کرد. بعلاوه ما همواره برای موقعیت‌هایی که جهت دستیابی به هر یک از سود مورد نظر یا ماکزیمم زیان مجاز، منتظر خواهیم بود.

ما تنها این دو استراتژی عمده تجاری را با انحرافات ناچیزی روی پارامترهای استفاده شده بررسی خواهیم کرد (برای مثال، زمان نگهداری، حاشیه سود مورد انتظار، یا مقدار پول سرمایه‌گذاری شده در هر موقعیت). همانطور که اشاره شد، این‌ها به سادگی برای اهداف شرح داده شده انتخاب شدند.

3-5-2- معیار ارزیابی مربوط به تجارت (فارکس)

استاندارد شرح داده شده در بخش 3-3-4 مستقیماً به هدف کلی این برنامه که با عملکرد اقتصادی انجام می‌شود، ترجمه نمی‌شود. فاکتورهای مانند نتایج اقتصادی و ریسک افشای بعضی ابزار یا دستورالعمل‌های مالی کلید مهمی در این متن هستند. این ناحیه‌ای است که به تنهایی می‌تواند به سادگی این فصل را پر کند. عملکرد بسیاری از استانداردهای مالی موجود برای تجزیه و (Carl and Peterson, 2009) تجزیه و تحلیلی پکیج تحلیل بازده بعضی الگوریتم‌های تجاری همچون آنکه ما در این فصل پیشنهاد دادیم را اجرا می‌کند. ما بعضی از

این توابع را بوسیله این پکیج برای جمع‌آوری اطلاعاتی در عملکرد اقتصادی پیشنهاداتمان استفاده خواهیم کرد. ارزیابی ما بر نتایج کلی مدل‌ها، ریسک افشای‌شان و نتایج میانگین هر کدام از موقعیت‌های نگه‌داشته شده توسط مدل، تمرکز دارد. در ارزیابی نهایی از سیستم پیشنهادی‌مان که در بخش 3-7 توضیح داده شد، ما یک تجزیه و تحلیل عمیق‌تری از عملکرد آن با استفاده از ابزارهای استفاده شده که در این کتاب فراهم شده است اجرا خواهیم کرد.

با احترام به نتایج کلی، ما استفاده خواهیم کرد: (1) توازن ساده خالصی بین سرمایه اولیه و سرمایه پایان دوره آزمایش (گاهی اوقات به نام سود/زیان نامیده می‌شود)، (2) درصد بازگشتی که این توازن خالص ارائه می‌کند و (3) بازده مازاد بیش از استراتژی خرید و نگهداری. این استراتژی شامل موقعیت‌های طولانی مدت بازگشایی در ابتدای دوره آزمایشی و منتظر ماندن تا وقتی که در پایان این بسته شود. بازده بیش از خرید و نگهداری بوسیله تفاوت میان بازگشت از استراتژی تجاری و این استراتژی ساده اندازه‌گیری می‌شود.

راجع به اندازه‌گیری ریسک مربوطه، ما از نسبت ضریب شارپ استفاده خواهیم کرد، که ریسک بازده هر واحد را اندازه‌گیری می‌کند، دومی همچون انحراف استاندارد از بازده اندازه‌گیری می‌شود. ما همچنین حداکثر کاهش نیروها را محاسبه خواهیم کرد که زیان انباشته متوالی یک مدل را اندازه‌گیری می‌کند.

این یک ریسک مهم اندازه‌گیری برای بازرگانان است، مانند یک سیستم که یک ترسیم رو به پایین جدی را چک می‌کند، احتمالاً در حالتی که برای اجرا بدون پول بماند محکوم به فنا است، مانند سرمایه‌گذارانی که مطمئناً با این زیان‌های پی‌درپی و بیرون کشیدن مجدد پولشان خواهند ترسید. در نهایت، عملکرد موقعیت نگهداری در خلال دوره آزمایشی با اعدادشان، میانگین بازده در هر موقعیت و درصد موقعیت‌های سوددهی به خوبی سایر استانداردهای مربوطه زیان‌دهی ارزیابی خواهد شد.

3-5-3- قرار دادن همه چیز با هم: یک بازرگان شبیه‌سازی شده

این بخش توضیح می‌دهد چگونه نظریه‌ای را که در رابطه با علامت‌ها و مدل‌های بازرگانی طراحی کردیم را فراهم می‌کند، که می‌تواند برای کنار هم گذاشتن (`trading.simulator()`) اجرا کنیم. پکیج کتاب ما تابع تمام این نظرات بوسیله یک شبیه‌سازی تجاری که از علامت‌های هر مدل ایجاد می‌شود، استفاده گردد. پارامترهای اصلی این بخش قیمت جاری بازار برای هر دوره شبیه‌سازی و علامت‌های مدل برای این دوره است. دو پارامتر دیگر نام تابع خطی‌مشی تعریف شده توسط کاربر و لیستش از پارامترها می‌باشد. در نهایت، ما همچنین می‌توانیم قیمت هر مبادله و سرمایه اولیه در دسترس برای هر بازرگان را مشخص کنیم. شبیه‌ساز تابع سیاست تجاری فراهم شده توسط هر کاربر در پایان هر بخش روزانه نامیده می‌شود و تابع باید سفارشات را که شبیه‌ساز برای اجرا کردن می‌خواهد، بازگرداند. شبیه‌ساز این سفارشات را در بازار انجام می‌دهد و تما `tradeRecord` فعالیت‌های در مورد چندین ساختار اطلاعات را ثبت می‌کند. نتایج شبیه‌سازی هدف کلاس می‌باشد که شامل اطلاعات این شبیه‌سازی است. این هدف، همچنان که خواهیم دید می‌تواند سپس در توابع دیگری برای بدست آوردن استاندارد یا نمودارهای ارزیابی اقتصادی در فعالیت‌های تجاری استفاده شود. قبل از پردازش بوسیله یک مثال از این گونه شبیه‌سازی ما به فراهم کردن جزئیات آینده در باره توابع خط‌مشی

سیاسی نیاز داریم که کاربر برای تدارک دیدن این شبیه‌ساز لازم دارد. این توابع باید با استفاده از یک پروتکل معین نوشته شود که آن‌ها باید از اینکه چگونه شبیه‌ساز آن‌ها را فرا خواهد خواند، آگاه باشند و باید اطلاعات، شبیه‌ساز تابع خط‌مشی تجاری را با `id` این شبیه‌ساز که مورد انتظار است را بازگرداند. در پایان هر بخش روزانه چهار آرگومان اصلی به اضافه هر پارامتر دیگر که مصرف کننده در فراخوان این شبیه‌سازی استفاده کرده است، (2) قیمت جاری `d` را می‌خواند. این چهار آرگومان هستند: (1) یک بردار با واحد نشانه‌های پیش‌بینی روز (3) موقعیت‌های باز شده جاری و (4) پولی که اخیراً برای تجارت در دسترس است. همچنان که `d` بازار (تا روز وجود دارد، موقعیت اخیر یک ماتریکس با ردیف‌های بسیار است. این `d` آنجا موقعیت‌های باز در پایان روز که می‌تواند 1 برای موقعیت‌های طولانی مدت یا 1- برای `"pos.type"` ماتریکس چهار ستون دارد؛ که روزی است که `"Odate"` که تعداد سهامی در موقعیت است؛ `"N.stocks"` موقعیت‌های کوتاه مدت باشد؛ که قیمتی است که در آن موقعیت باز شده `"Oprice"`؛ و `d` آن موقعیت باز شده است (یک عدد بین 1 و از موقعیت‌هایی است که مربوط است زمانی که می‌خواهیم `IDs` است. نام‌های ردیف این ماتریکس شامل شبیه‌سازی را نشان دهیم که یک موقعیت قطعی بسته می‌شود.

تمام این اطلاعات بوسیله این شبیه‌ساز فراهم می‌شود تا اطمینان دهد که استفاده کننده می‌تواند یک مجموعه گسترده از خط‌مشی تجاری را تعریف کند. توابع تعریف شده توسط کاربر باید یک چارچوب اطلاعات با یک گروه از سفارشات که شبیه‌ساز باید اجرا کند را بازگرداند این چارچوب اطلاعات باید شامل که باید 1 برای سفارشات خرید و 1- برای سفارشات فروش باشد؛ `"order"` (ستون) اطلاعات زیرین باشد؛ که باید 1 برای سفارشات بازار که فوراً انجام می‌شوند (در حقیقت در چند روز آتی بازگشایی `"order.type"` که باید مقدار سهام مبادله شده برای `"val"` قیمت)، 2 برای سفارشات محدود یا 3 برای دستورات توقف باشد؛ برای دستورات بستن بازار یا یک قیمت هدف برای دستورات توقف یا `NA` دستورات بازگشایی بازار باشد، که باید برای دستوراتی که یک موقعیت جدید می‌گشاید باز باشد یا برای دستورات `"action"` محدودیت؛ یک موقعیت `ID` که اگر قابل اجراست باید شامل `"postD"` بستن یک موقعیت موجود بسته باشد؛ و در نهایت که بسته می‌شود باشد.

در زیر توضیحی از یک تابع خط‌مشی تجاری تعریف شده بر مبنای کاربر آمده است:

```
> policy.1 <- function(signals,market,opened.pos,money,
+                       bet=0.2,hold.time=10,
+                       exp.prof=0.025, max.loss= 0.05
+                       )
+ {
+   d <- NROW(market) # this is the ID of today
+   orders <- NULL
+   nOs <- NROW(opened.pos)
+   # nothing to do!
```

```

+   if (!nOs && signals[d] == 'h') return(orders)
+
+   # First lets check if we can open new positions
+   # i) long positions
+   if (signals[d] == 'b' && !nOs) {
+     quant <- round(bet*money/market[d,'Close'],0)
+     if (quant > 0)
+       orders <- rbind(orders,
+         data.frame(order=c(1,-1,-1),order.type=c(1,2,3),
+           val = c(quant,
+             market[d,'Close']*(1+exp.prof),
+             market[d,'Close']*(1-max.loss)
+           ),
+           action = c('open','close','close'),
+           posID = c(NA,NA,NA)
+         )
+       )
+
+   # ii) short positions
+   } else if (signals[d] == 's' && !nOs) {
+     # this is the nr of stocks we already need to buy
+     # because of currently opened short positions
+     need2buy <- sum(opened.pos[opened.pos[, 'pos.type'] == -1,
+       "N.stocks"])*market[d,'Close']
+     quant <- round(bet*(money-need2buy)/market[d,'Close'],0)
+     if (quant > 0)
+       orders <- rbind(orders,
+         data.frame(order=c(-1,1,1),order.type=c(1,2,3),
+           val = c(quant,
+             market[d,'Close']*(1-exp.prof),
+             market[d,'Close']*(1+max.loss)
+           ),
+
+           action = c('open','close','close'),
+           posID = c(NA,NA,NA)
+         )
+       )
+   }
+
+ }

```

```

+ # Now lets check if we need to close positions
+ # because their holding time is over
+ if (nOs)
+   for(i in 1:nOs) {
+     if (d - opened.pos[i,'Odate'] >= hold.time)
+       orders <- rbind(orders,
+         data.frame(order=-opened.pos[i,'pos.type'],
+           order.type=1,
+           val = NA,
+           action = 'close',
+           posID = rownames(opened.pos)[i]
+         )
+       )
+   }
+   orders
+ }

```

اولین استراتژی تجاری را که در بخش 3-5-1 شرح دادیم را اجرا می‌کند. تابع 4 پارامتر `policy.1()` تابع دارد که می‌توانیم برای سازگاری این استراتژی استفاده کنیم. این‌ها پارامتر شرط هستند که درصد پول جاری را `exp.prof` مشخص می‌کند، که ما هر زمانی یک موقعیت جدید باز کنیم، سرمایه‌گذاری خواهیم کرد؛ پارامتر که حاشیه سودی را نشان می‌دهد که آرزو داریم برای موقعیت‌هایمان بدست آوریم و استفاده از آن زمانی که که حداکثر زبانی را نشان می‌دهد که ما آرزو داریم آن را قبل از `max.loss` دستورات محدودیت پست می‌شود؛ ، که تعداد روزهایی را `hold.time` اینکه موقعیت را ببندیم بپذیریم، و استفاده آن در دستور توقف؛ و پارامتر نشان می‌دهد که ما آرزو داریم برای دستیابی به حاشیه سود منتظر بمانیم. اگر به زمان نگهداری بدون دستیابی به حاشیه خواسته شده دست یافتیم، موقعیت بسته می‌شود.

توجه داشته باشید که هر وقت ما یک موقعیت جدید باز کنیم، سه دستور به شبیه‌ساز ارسال می‌کنیم: یک دستور بازار برای باز کردن موقعیت جدید، یک دستور محدودیت برای مشخص کردن حاشیه سود هدفمان و دستور توقف برای محدود کردن زیان‌هایمان.

به طور هم‌ارز، تابع زیر استراتژی دوم تجاری‌مان را اجرا می‌کند:

```

> policy.2 <- function(signals,market,opened.pos,money,
+   bet=0.2,exp.prof=0.025, max.loss= 0.05
+ )
+ {
+   d <- NROW(market) # this is the ID of today

```

```

+ orders <- NULL
+ nOs <- NROW(opened.pos)
+ # nothing to do!
+ if (!nOs && signals[d] == 'h') return(orders)
+
+ # First lets check if we can open new positions
+ # i) long positions
+ if (signals[d] == 'b') {
+   quant <- round(bet*money/market[d,'Close'],0)
+   if (quant > 0)
+     orders <- rbind(orders,
+       data.frame(order=c(1,-1,-1),order.type=c(1,2,3),
+         val = c(quant,
+           market[d,'Close']*(1+exp.prof),
+           market[d,'Close']*(1-max.loss)
+         ),
+       action = c('open','close','close'),
+       posID = c(NA,NA,NA)
+     )
+   )
+
+ # ii) short positions
+ } else if (signals[d] == 's') {
+   # this is the money already committed to buy stocks
+   # because of currently opened short positions
+   need2buy <- sum(opened.pos[opened.pos[, 'pos.type'] == -1,
+     "N.stocks"])*market[d,'Close']
+   quant <- round(bet*(money-need2buy)/market[d,'Close'],0)
+   if (quant > 0)
+     orders <- rbind(orders,
+       data.frame(order=c(-1,1,1),order.type=c(1,2,3),
+         val = c(quant,
+           market[d,'Close']*(1-exp.prof),
+           market[d,'Close']*(1+max.loss)
+         ),
+       action = c('open','close','close'),
+       posID = c(NA,NA,NA)
+     )
+   )
+ }

```



```
+
+   orders
+ }
```

این تابع بسیار شبیه تابع قبلی است. تفاوت اصلی بر این حقیقت تکیه دارد که در این خط‌مشی سیاسی ما برای موقعیت‌های بیشتری اجازه می‌دهیم که در همان زمان بازگشایی شود و همچنین اینجا هیچ محدودیت زمانی برای موقعیت‌های بسته شدن وجود ندارد.

با داشتن تعریفی از خط‌مشی تجاری توابع، ما برای درست کردن شبیه‌ساز تجاری مان آماده هستیم. برای انتخاب خواهیم کرد که SVM توضیح مقصودمان یک نمونه کوچکی از اطلاعاتمان را برای دستیابی به یک سپس برای دستیابی به پیش‌بینی‌هایی برای یک دوره ثانویه استفاده می‌شود. ما شبیه‌ساز تجاریمان را با این در متن از خط‌مشی معین تجاری SVM پیش‌بینی‌ها برای دستیابی به نتایج تجاری با استفاده از علامت‌های نامگذاری می‌کنیم.

```
> # Train and test periods
> start <- 1
> len.tr <- 1000
> len.ts <- 500
> tr <- start:(start+len.tr-1)
> ts <- (start+len.tr):(start+len.tr+len.ts-1)
> # getting the quotes for the testing period
> data(GSPC)
> date <- rownames(Tdata.train[start+len.tr,])
> market <- GSPC[paste(date,'/',sep=")")[1:len.ts]
> # learning the model and obtaining its signal predictions
> library(e1071)
> s <- svm(Tform, Tdata.train[tr,], cost=10, gamma=0.01)
> p <- predict(s, Tdata.train[ts,])
> sig <- trading.signals(p, 0.1, -0.1)
> # now using the simulated trader
> t1 <- trading.simulator(market, sig,
+   'policy.1', list(exp.prof=0.05, bet=0.2, hold.time=30))
```

لطفا توجه داشته باشید برای اینکه این کد کار کند، شما باید قبلاً هدف‌هایی با اطلاعات برای مدل‌سازی با استفاده از دستورالعمل‌های داده شده در بخش 3-3-3 خلق کنید.

در فراخوان خود برای شبیه سازی تجاری ما اولین خطمشی تجاری را انتخاب کردیم و ارزش های متفاوتی را برای برخی از این پارامترها فراهم کردیم. ما از اطلاعات پیش فرض برای هزینه های مبالغه (پنج واحد پولی) و برای سرمایه اولیه (1 میلیون واحد پولی) استفاده کردیم. نتیجه این فراخوانی یک شیء از کلاس tradeRecord است. ما می توانیم این مفهوم را همچون زیر چک کنیم:

> t1

Object of class tradeRecord with slots:

```
trading: <xts object with a numeric 500 x 5 matrix>
positions: <numeric 16 x 7 matrix>
init.cap : 1e+06
trans.cost : 5
policy.func : policy.1
policy.pars : <list with 3 elements>
> summary(t1)
```

== Summary of a Trading Simulation with 500 days ==

Trading policy function : policy.1

Policy function parameters:

exp.prof = 0.05

bet = 0.2

hold.time = 30

Transaction costs : 5

Initial Equity : 1e+06

Final Equity : 997211.9 Return : -0.28 %

Number of trading positions: 16

برای آیتم های عددی آتی در این شبیه سازی استفاده شد. تابع tradingEvaluation() تابع می تواند برای بدست آوردن یک سری از نشانه های اقتصادی عملکرد در خلال این tradingEvaluation() دوره شبیه سازی استفاده گردد:

> tradingEvaluation(t1)

NTrades	NProf	PercProf	PL	Ret	RetOverBH
---------	-------	----------	----	-----	-----------

16.00	8.00	50.00	-2788.09	-0.28	-7.13	
MaxDD	SharpeRatio		AvgProf	AvgLoss	AvgPL	MaxProf
59693.15	0.00	4.97	-4.91	0.03	5.26	
MaxLoss						
-5.00						

یک مطالعه کلی گرافیکی از عملکرد بازرگان `plot()` ما مانند زیر همچنین می‌توانیم با استفاده از تابع بدست بیاوریم:

```
> plot(t1, market, theme = "white", name = "SP500")
```

نتیجه این فرمان در نمودار 6.3 نشان داده شده است. با یک بازده منفی، نتایج این تجارت بد است. اگر ما خط‌مشی تجاری ثانویه‌ای استفاده کنیم آیا سناریو متفاوت می‌شود؟ اجازه دهید ببینیم:

```
> t2 <- trading.simulator(market, sig, "policy.2", list(exp.prof = 0.05,
+   bet = 0.3))
> summary(t2)
```

== Summary of a Trading Simulation with 500 days ==

Trading policy function : policy.2

Policy function parameters:

exp.prof = 0.05

bet = 0.3

Transaction costs : 5

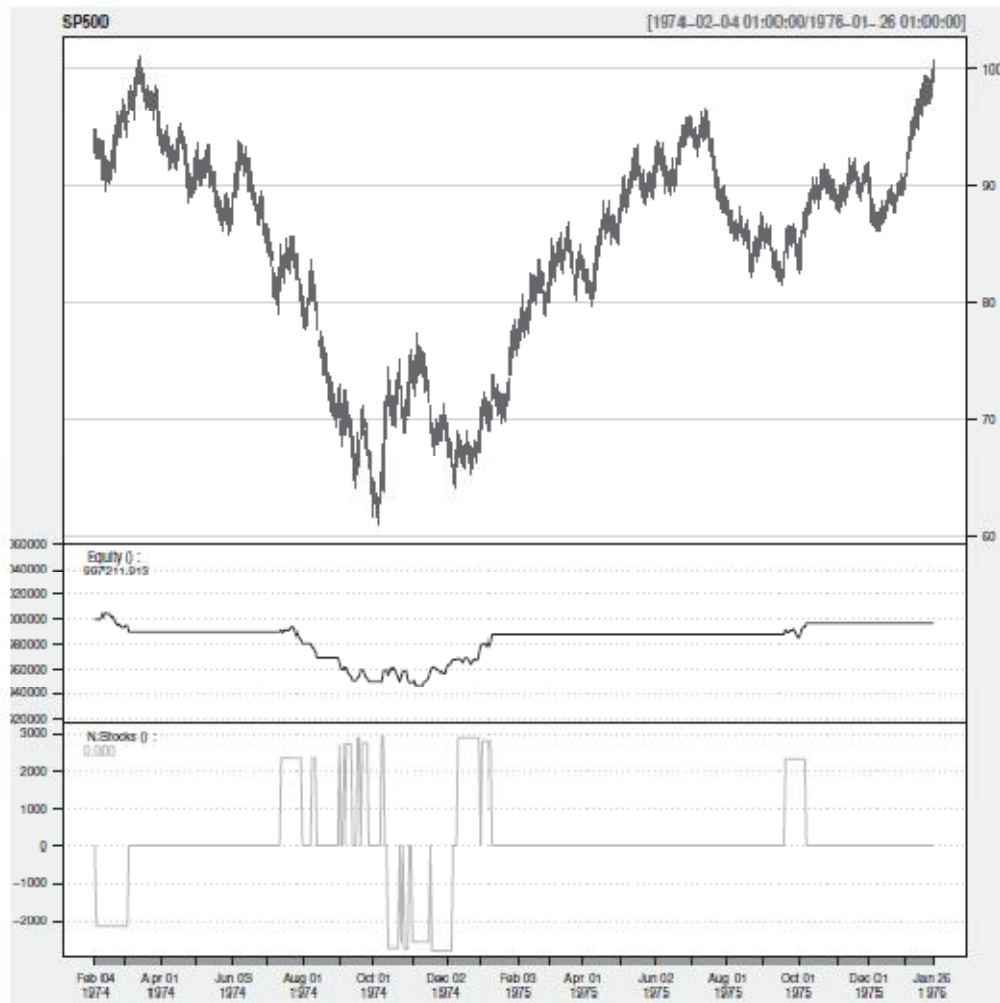
Initial Equity : 1e+06

Final Equity : 961552.5 Return : -3.84 %

Number of trading positions: 29

را برای آیتم‌های عددی آتی در این شبیه‌سازی استفاده کنید. `tradingEvaluation()` تابع

```
> tradingEvaluation(t2)
```



SVM نمودار 3-6: نتایج مبادله با استفاده از خط‌مشی 1 بر مبنای علامت‌هایی از یک

NTrades	NProf	PercProf	PL	Ret	RetOverBH	
29.00	14.00	48.28	-38447.49	-3.84	-10.69	
MaxDD	SharpeRatio		AvgProf	AvgLoss	AvgPL	MaxProf
156535.05	-0.02		4.99	-4.84	-0.10	5.26
MaxLoss						
-5.00						

از همان علامت‌ها استفاده شده با یک خط‌مشی تجاری متفاوت بازده از 0/27% تا 2/86%- کاهش می‌یابد.
به ما اجازه دهید تا تجربه را با آموزش و دوره آزمون متفاوت تجربه کنیم:

```
> start <- 2000
> len.tr <- 1000
```

```

> len.ts <- 500
> tr <- start:(start + len.tr - 1)
> ts <- (start + len.tr):(start + len.tr + len.ts - 1)
> s <- svm(Tform, Tdata.train[tr, ], cost = 10, gamma = 0.01)
> p <- predict(s, Tdata.train[ts, ])
> sig <- trading.signals(p, 0.1, -0.1)
> t2 <- trading.simulator(market, sig, "policy.2", list(exp.prof = 0.05,
+      bet = 0.3))
> summary(t2)

```

== Summary of a Trading Simulation with 500 days ==

Trading policy function : policy.2

Policy function parameters:

exp.prof = 0.05

bet = 0.3

Transaction costs : 5

Initial Equity : 1e+06

Final Equity : 107376.3 Return : -89.26 %

Number of trading positions: 229

برای آیتم‌های عددی آتی در این شبیه‌سازی استفاده کنید `tradingEvaluation()` از تابع

```

> tradingEvaluation(t2)

```

NTrades	NProf	PercProf	PL	Ret	RetOverBH
229.00	67.00	29.26	-892623.73	-89.26	-
96.11					
MaxDD	SharpeRatio		AvgProf	AvgLoss	AvgPL
MaxProf					
959624.80	-0.08	5.26	-4.50	-1.65	5.26
MaxLoss					
-5.90					

این بازرگان، با همان تکنیک مدل‌سازی و استفاده از همان استراتژی تجاری بدست آمده است، یک نتیجه بدتر مورد توجهی بدست آمده است. درس اصلی برای یادگیری این است: تخمین‌های آماری معتبر، با چند تکرار آزمایش‌ها فریب نخورید حتی اگر این شامل 2 سال دوره آزمایشی باشد. ما تکرارهای بیشتری تحت شرایط

متفاوت برای اطمینان یافتن از قابلیت اطمینان آماری بعضی نتایج مان نیاز داریم. این به طور ویژه برای مدل‌های سری زمانی صحیح است که رژیم‌های متفاوتی برای مدیریت کردن داشته باشد (برای مثال، دوره‌هایی با گرایش یا فراریت متفاوت)

3-6- ارزیابی مدل و انتخاب

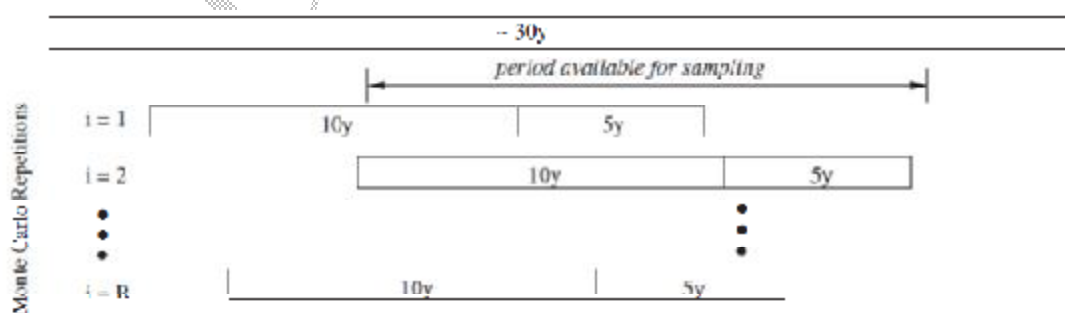
در این بخش ما بررسی خواهیم کرد چگونه یک تخمین قابل اعتماد از معیار ارزیابی انتخاب شده را بدست آوریم. این تخمین‌ها به ما اجازه خواهد داد که بطور کامل سیستم‌های تجاری متناوب مختلف را مقایسه و انتخاب کنیم.

3-6-1- برآوردهای مونت کارلو

مشکلات سری زمانی مانند آنکه ما اشاره کردیم چالش‌های جدیدی در اصطلاح دستیابی به تخمین‌های قابل اعتماد از استاندارد اندازه‌گیری ایجاد می‌کند. این به دلیل است که همه اطلاعات مشاهده شده یک برچسب زمانی متصل دارند که یک سفارش را میان آن‌ها لازم می‌سازد. این سفارشات باید با ریسک برآوردهای بدست آمده که قابل اتکا نیست مربوط باشد. در فصل 2 ما از روش "وارسی اعتبار" برای بدست آوردن برآورد قابل اتکایی از آمارهای تخمینی استفاده کردیم. این روش شامل یک مرحله نمونه‌گیری تصادفی مجدد است که دستورات اصلی از مشاهدات را تغییر می‌دهد. این بدین معنی است که واری اعتبار نباید به مشکلات سری‌های زمانی بکار رود. کاربرد این روش می‌تواند به معنی آزمایش مدل‌ها بر مشاهداتی که قدیمی‌تر از آنهایی که برای بدست آوردن آن‌ها استفاده شد باشد. این در واقعیت امکان‌پذیر نیست و بنابراین تخمین‌های بدست آمده با این فرایند غیرقابل اتکا و احتمالاً بسیار خوشبینانه هستند، همچنانکه این آسانتر است گذشته داده شده را پیش‌بینی کنیم تا اینکه آینده پیش رو. هر فرایند برآورد کردنی که از داده سری زمانی استفاده می‌کند باید اطمینان دهد که مدل‌ها همیشه در تاریخی تست شده‌اند که بسیار جدیدتر از اطلاعات استفاده شده برای بدست آوردن مدل‌ها هستند. این یعنی هیچ نمونه‌گیری تصادفی مجددی از مشاهدات یا هیچ فرایند دیگری که دستورات زمانی از اطلاعات داده شده را تغییر دهد، وجود ندارد. اگرچه هر فرایند تخمین مناسب باید شامل بعضی انتخابات تصادفی برای اطمینان یافتن از قابلیت اتکای آماری هر تخمین بدست آمده باشد. این شامل تکرار کردن چندین باره فرایند تخمین تحت شرایط متفاوت، با انتخاب ترجیحاً تصادفی، است. اطلاعات یک اندازه‌گیری می‌شود، چگونه می‌تواند ما را به این امر مطمئن $t+N$ تا زمان t سری زمانی داده شده که از زمان را برای آنچه می‌خواهیم تخمینش را به دست آوریم، انتخاب $train+test$ سازد؟ اول اینکه، ما باید تنظیمات کنیم. این یعنی تصمیم‌گیری اینکه اندازه هر دو گروه آزمایشی و آموزشی برای استفاده شدن در فرایند تخمین، باشد تا اطمینان دهد که ما قادریم به طور تصادفی N چقدر است؟ مجموع این دو اندازه باید کوچکتر از سناریوهای آزمایشی مختلفی را با اطلاعاتی که بوسیله ما فراهم شده است را تولید کنیم. اگرچه، در صورتیکه ما اندازه‌های بیش از حد کوچکی را انتخاب کنیم ممکن است عملکرد مدل‌هایمان بطور جدی معیوب شود. بهمین نحو، بسته‌های آزمایشی کوچک کمتر قابل اتکا خواهند بود بویژه اگر ما مظنون باشیم که آنجا در مدل‌مان رژیمی وجود دارد که تغییر کرده است و ما تمایل داشته باشیم مدل‌ها را تحت همین شرایط آزمایش کنیم.

تقریباً بسته اطلاعاتی ما شامل 30 سال قیمت‌های جاری روزانه است. و قتیکه 10 سال اطلاعات آموزشی وجود دارد، ما تمام تناوبات را بوسیله تخمین عملکردشان در آزمایش گروه 5 ساله قیمت‌های جاری ارزیابی خواهیم کرد. این اندازه‌های آزمایشی و آموزشی را قابل اطمینان می‌سازد که به قدر کافی بزرگ هستند و علاوه بر این، فضایی برای تکرارهای متفاوت این فرایند آزمایشی همچنان که در اطلاعات 30 ساله داریم، باقی می‌گذارد.

از لحاظ روش تجربی، ما تجربه مونت کارلو را برای بدست آوردن تخمین‌های قابل اتکا از استانداردهای ارزیابی استفاده خواهیم کرد. روش مونت کارلو بر نمونه‌گیری تصادفی برای رسیدن به نتایج‌شان تکیه دارد. ما در دوره زمانی 30 ساله R قصد داریم این روش نمونه‌گیری فرایند را برای انتخاب یک گروه از نقاط ، ما از دوره 10 ساله قبلی قیمت‌های جاری قیمت‌هایمان استفاده کنیم. برای هر انتخاب تصادفی نقطه زمانی برای بدست آوردن مدل‌ها و دوره 5 ساله متعاقب برای آزمایش آن‌ها استفاده خواهیم کرد. در پایان این تخمینی برای هر یک از استانداردهای ارزیابی مان داریم. هر کدام از این تخمین‌ها برای بدست R ما R تکرارهای آوردن یک انتخاب تصادفی پنجره از 15 سال اطلاعات است، اولین 10 سال برای آموزش و 5 سال باقیمانده جهت آزمایش استفاده می‌شود. این اطمینان می‌بخشد که تجربیات ما همیشه به سفارشات زمانی اطلاعات سری را اطمینان می‌دهد که train+test تغییر پذیری مناسب در شرایط R زمانی مرتبط است. تکرار فرایند زمانی تصادفی انتخاب شده را R قابل اتکا بودن تخمین‌هایمان را افزایش می‌دهد. بعلاوه اگر ما همان نقطه‌های مقدار برای تناوبات مختلف ارزیابی استفاده کنیم، می‌توانیم برای بدست آوردن سطح اعتماد آماری بر روی مشاهدات مختلف از عملکرد متوسط، مقایسه‌های جفتی انجام دهیم. نمودار 3-7 روش تجربی مونت کارلو را به طور ، 10 سال خلاصه نشان داده است. توجه داشته باشید که باید اطمینان یابیم که برای هر نقطه تصادفی حذف می‌کند. R اطلاعات قبل از 5 سال بعدی وجود دارد، این بعضی از داده‌ها را از انتخاب تصادفی نقاط



نمودار 3-7: فرایند تجربی مونت کارلو

k-، که در فصل 2 برای بدست آوردن آزمایشات واریسی اعتبار $\text{experimentalComparison}()$ تابع استفاده می‌کردیم، می‌تواند برای آزمایشات مونت کارلو نیز استفاده شود. در بخش بعدی ما این را برای fold

بدست آوردن تخمین‌های قابل اتکا از استانداردهای ارزیابی انتخاب شده برای چندین سیستم مبادلاتی متناوب استفاده خواهیم کرد.

3-6-2_ مقایسات تجربی

این بخش یک گروه از آزمایشات مونت کارلو را شرح می‌دهد که برای بدست آوردن تخمین‌های قابل اتکا از معیار ارزیابی اشاره شده در بخش 3-3 و 3-5 طراحی شده است. اطلاعات مبنایی استفاده شده در این تجربیات از دیتابسی که در پایان بخش 3-3 ایجاد شد، استفاده شده است. هر کدام از مدل‌های پیش‌بینی متناوب بررسی شده در این آزمایشات در تنظیمات به روز رسانی سه مدل مختلف استفاده خواهد شد. قبلاً این‌ها در بخش 3-4 شرح داده شده است و مشتمل بر استفاده از یک مدل واحد برای تمام دوره‌های آزمایشی 5 ساله، با استفاده از یک پنجره کشویی یا یک پنجره روینده، می‌باشد. پکیج کتاب شامل 2 تابع است که به `slidingWindow()` و `growingWindow()` استفاده از هر مدلی با این تدبیر ایجاد پنجره، کمک می‌کند. تابع 5 آرگومان اصلی دارد. اولی یک شیء از کلاس آموزشی است که قبلاً برای نگهداری تمام جزئیات در یک سیستم آموزشی از آن استفاده کردیم (تابع نام و پارامتر ارزش (بها)). دومین آرگومان فرمولی است که وظیفه پیش‌بینی را شرح می‌دهد، در حالیکه سومین و چهارمین شامل آموزش و آزمایش گروه داده‌های مربوطه است. آرگومان آخر تنظیمات آموزش مجدد برای استفاده در تدبیر ایجاد پنجره است. پس از آنکه تعداد موارد تست شده در این آرگومان مشخص شد، به هر کدام از شیوه‌های کشویی یا روینده اطلاعات آموزشی استفاده شده برای بدست آوردن مدل قبلی، مدل بازآموزی می‌شود. هر دو تابع پیش‌بینی‌ها از مدل را برای فراهم کردن گروه تست استفاده شده در تدبیر ایجاد پنجره مربوطه بازمی‌گرداند.

کدهای زیر گروهی از توابع که برای اجرای یک چرخه کامل آموزش+تست+ارزیابی از سیستم‌های مبادلاتی مختلف استفاده خواهد کرد، را ایجاد می‌کند که ما آن‌ها را مقایسه خواهیم نمود. این توابع از درون روال مونت کارلو برای دوره‌های مختلف آموزش+تست طبق طرح توصیف شده در نمودار 3-7 نام‌گذاری خواهد شد.

```
> MC.svmR <- function(form, train, test, b.t = 0.1, s.t = -0.1,
+ ...) {
+   require(e1071)
+   t <- svm(form, train, ...)
+   p <- predict(t, test)
+   trading.signals(p, b.t, s.t)
+ }
> MC.svmC <- function(form, train, test, b.t = 0.1, s.t = -0.1,
+ ...) {
+   require(e1071)
+   tgtName <- all.vars(form)[1]
+   train[, tgtName] <- trading.signals(train[, tgtName],
+   b.t, s.t)
```



```

+   t <- svm(form, train, ...)
+   p <- predict(t, test)
+   factor(p, levels = c("s", "h", "b"))
+ }
> MC.nnetR <- function(form, train, test, b.t = 0.1, s.t = -0.1,
+ ...) {
+   require(nnet)
+   t <- nnet(form, train, ...)
+   p <- predict(t, test)
+   trading.signals(p, b.t, s.t)
+ }
> MC.nnetC <- function(form, train, test, b.t = 0.1, s.t = -0.1,
+ ...) {
+   require(nnet)
+   tgtName <- all.vars(form)[1]
+   train[, tgtName] <- trading.signals(train[, tgtName],
+   b.t, s.t)
+   t <- nnet(form, train, ...)
+   p <- predict(t, test, type = "class")
+   factor(p, levels = c("s", "h", "b"))
+ }
> MC.earth <- function(form, train, test, b.t = 0.1, s.t = -0.1,
+ ...) {
+   require(earth)
+   t <- earth(form, train, ...)
+   p <- predict(t, test)
+   trading.signals(p, b.t, s.t)
+ }
> single <- function(form, train, test, learner, policy.func,
+ ...) {
+   p <- do.call(paste("MC", learner, sep = "."), list(form,
+   train, test, ...))
+   eval.stats(form, train, test, p, policy.func = policy.func)
+ }
> slide <- function(form, train, test, learner, relearn.step,
+ policy.func, ...) {
+   real.learner <- learner(paste("MC", learner, sep = "."),
+   pars = list(...))
+   p <- slidingWindowTest(real.learner, form, train, test,
+   relearn.step)

```

```

+   p <- factor(p, levels = 1:3, labels = c("s", "h", "b"))
+   eval.stats(form, train, test, p, policy.func = policy.func)
+ }
> grow <- function(form, train, test, learner, relearn.step,
+   policy.func, ...) {
+   real.learner <- learner(paste("MC", learner, sep = "."),
+     pars = list(...))
+   p <- growingWindowTest(real.learner, form, train, test,
+     relearn.step)
+   p <- factor(p, levels = 1:3, labels = c("s", "h", "b"))
+   eval.stats(form, train, test, p, policy.func = policy.func)
+ }

```

مدل‌های متفاوتی را با استفاده از فرمول فراهم شده و گروه آموزشی بدست می‌آورد و سپس `MC.x()` تابع آن‌ها را در گروه آزمایشی داده شده جهت بازده پیش‌بینی‌ها، تست می‌کند. در زمان مقتضی، ما یک نسخه برای پایان نامیده `C` پایان نامیده می‌شود) و نسخه دیگری برای وظیفه طبقه‌بندی (در `R` وظیفه رگرسیون (در می‌شود) داریم. توجه داشته باشید که هر دوی این تناوبات فرایندهای "قبل" و "بعد" متفاوتی را دنبال نامیده می‌شوند. این سه تابع پیش‌بینی‌هایی `grow()`، `Slide()`، `single()` می‌کنند. این توابع به عنوان توابع برای گروه تست استفاده شده در مدل خاصی در پارامترهای آموزنده با استفاده از مکانیسم به روز رسانی مدل مربوطه، بدست می‌آورد. پس از بدست آمدن پیش‌بینی‌ها این توابع آمارهای ارزیابی را جمع‌آوری می‌کند که که در زیر داده شده است را ارزیابی کنیم. `eval.stats()` می‌خواهیم با یک نامگذاری برای تابع

```

> eval.stats <- function(form, train, test, preds, b.t=0.1, s.t=-0.1, ...) {
+   # Signals evaluation
+   tgtName <- all.vars(form)[1]
+   test[,tgtName] <- trading.signals(test[,tgtName], b.t, s.t)
+   st <- sigs.PR(preds, test[,tgtName])
+   dim(st) <- NULL
+   names(st) <- paste(rep(c('prec', 'rec'), each=3),
+     c('s', 'b', 'sb'), sep='.')
+
+   # Trading evaluation
+   date <- rownames(test)[1]
+   market <- GSPC[paste(date, "/", sep=")"]][1:length(preds),]
+   trade.res <- trading.simulator(market, preds, ...)
+
+   c(st, tradingEvaluation(trade.res))
+ }

```

+ }

دو تابع دیگر را برای جمع‌آوری دقت و فراخوانی علامت‌ها و چندین استاندارد ارزیابی `eval.stats()` تابع بعنوان مثال آرگومان‌های پیش‌بینی و درستی اطلاعات را دریافت `sigs.PR()` اقتصادی، استفاده می‌کند. تابع می‌کند و صحت و فراخوانی را برای علامت‌های فروش، خرید و خرید+فروش محاسبه می‌کند. تابع دیگر است که استانداردهای اقتصادی از ثبت مبادلات داده شده را بدست می‌آورد. این `tradingEvaluation()` بدست آمده است که می‌تواند به عنوان عملکرد شبیه‌سازی در `trading.simulator()` ثبت مبادلات با تابع بازار با علامت‌های مدل، استفاده شود. تمام این توابع به طور کامل در بخش 3-5-3 توضیح داده شده و با مثال از روال مونت کارلو با پارامترهای مناسب پر شده، `grow()` و `slide()`، `single()` مشخص گردیده است. توابع نامگذاری گردیده است بنابراین ما مدل‌های که می‌خواهیم مقایسه کنیم را بدست می‌آوریم. در زیر ما شرح می‌دهیم چگونه یک دور گردش (حلقه) را تنظیم کنیم که فراتر از یک گروه از سیستم‌های مبادلاتی متناوب باشد و این توابع را برای بدست آوردن تخمین عملکردشان نامگذاری کنیم. هر کدام از سیستم‌های مبادلاتش بوسیله مدل آموزشی با برخی پارامترهای آموزشی خاص شکل گرفته است، بعلاوه یک استراتژی تجاری که مشخص می‌شود این است که چگونه پیش‌بینی‌های مدل برای مبادله استفاده می‌شود. با احترام به خطی‌مشی‌های تجاری، ما سه گونه که از سیاست‌های خاصی در بخش 3-5-3 نتیجه می‌شود را بررسی خواهیم کرد. توابع ذیل این سه گونه را اجرا می‌کند: `policy.2()` و `policy.1()` کرد (تابع)

```
> pol1 <- function(signals,market,op,money)
+   policy.1(signals,market,op,money,
+           bet=0.2,exp.prof=0.025,max.loss=0.05,hold.time=10)
> pol2 <- function(signals,market,op,money)
+   policy.1(signals,market,op,money,
+           bet=0.2,exp.prof=0.05,max.loss=0.05,hold.time=20)
> pol3 <- function(signals,market,op,money)
+   policy.2(signals,market,op,money,
+           bet=0.5,exp.prof=0.05,max.loss=0.05)
```

کدهای زیر در آزمایشات مونت کارلو اجرا می‌شود. ما توصیه می‌کنیم که شما قبل از اجرای این کد چندین بار آن را بررسی کنید. حتی در سریع‌ترین کامپیوترها، این کار برای تکمیل شدن بع چندین روز زمان نیاز دارد. ما در وب پیج کتاب اهداف نتیجه را از اجرای این آزمایشات را تهیه کرده‌ایم بنابراین شما می‌توانید به تجزیه و تحلیل نتایجی که در ادامه می‌آید حتی بدون اجرا کردن این آزمایش روی کامپیوتر خودتان اتکا کنید.

```
> # The list of learners we will use
> TODO <- c('svmR','svmC','earth','nnetR','nnetC')
> # The datasets used in the comparison
```

```

> DSs <- list(dataset(Tform,Tdata.train,'SP500'))
> # Monte Carlo (MC) settings used
> MCsetts <- mcSettings(20, # 20 repetitions of the MC exps
+                       2540, # ~ 10 years for training
+                       1270, # ~ 5 years for testing
+                       1234) # random number generator seed
> # Variants to try for all learners
> VARS <- list()
> VARS$svmR <- list(cost=c(10,150),gamma=c(0.01,0.001),
+                  policy.func=c('pol1','pol2','pol3'))
> VARS$svmC <- list(cost=c(10,150),gamma=c(0.01,0.001),
+                  policy.func=c('pol1','pol2','pol3'))
> VARS$earth <- list(nk=c(10,17),degree=c(1,2),thresh=c(0.01,0.001),
+                  policy.func=c('pol1','pol2','pol3'))
> VARS$nnetR <- list(linout=T,maxit=750,size=c(5,10),
+                  decay=c(0.001,0.01),
+                  policy.func=c('pol1','pol2','pol3'))
> VARS$nnetC <- list(maxit=750,size=c(5,10),decay=c(0.001,0.01),
+                  policy.func=c('pol1','pol2','pol3'))
> # main loop
> for(td in TODO) {
+   assign(td,
+         experimentalComparison(
+           DSs,
+           c(
+             do.call('variants',
+                   c(list('single',learner=td),VARS[[td]],
+                     varsRootName=paste('single',td,sep='.'))),
+             do.call('variants',
+                   c(list('slide',learner=td,
+                         relearn.step=c(60,120)),
+                     VARS[[td]],
+                     varsRootName=paste('slide',td,sep='.'))),
+             do.call('variants',
+                   c(list('grow',learner=td,
+                         relearn.step=c(60,120)),
+                     VARS[[td]],
+                     varsRootName=paste('single',td,sep='.')))
+           ),
+         MCsetts)

```

```

+ )
+
+ # save the results
+ save(list=td,file=paste(td,'Rdata',sep='.'))
+ }

```

پارامترهای عمومی از آزمایش را که تعداد تکرارها (20)، اندازه گروه آزمایشی (تقریباً بین 2/540 تا 10 سال)، اندازه گروه آزمایشی (تقریباً بین 1/270 تا 5 سال) و تعداد تصادفی بذر تولید شده را مشخص می‌سازد، کنترل می‌کند.

شامل تمام پارامترهای مختلفی است که می‌خواهیم برای هر یادگیرنده بسنجیم. متغیرها VARS لیست شامل همه ترکیب‌های ممکن از ارزش‌ها که در لیست پارامترها نشان دادیم، می‌باشد. هر کدام از این متغیرها اجرا خواهد شد: پنجره تک، کشویی و روینده. بعلاوه، برای دو "modes" در سه مدل مختلف به روز رسانی شیوه بعدی دو مرحله بازآموزی را امتحان خواهیم کرد: دوره‌های 60 و 120 روزه.

ما سعی داریم چهار متغیر پارامتر را با سه سیاست مبادله مختلف که در کل 12 متغیر svm برای مدل‌های می‌شود را یاد بگیریم. هر کدام از این متغیرها در یک شیوه تکی سنجیده می‌شود و روی 4 تدبیر ایجاد پنجره (دو استراتژی با دو مرحله متفاوت بازآموزی). این امر به طور آشکارا در آزمایشات بسیاری که اجرا شده است (120=24+24+24)، 120 متغیر زمین (24+48+48=) و 60 svm نتیجه می‌دهد. به طور مثال، آنجا 60 متغیر خواهد بود. هر کدام از آن‌ها 20 مرتبه با یک گروه آموزشی 10 ساله و یک گروه تست 5 ساله اجرا nnet متغیر خواهد شد. این دلیلی است که ما خاطر نشان می‌کنیم که زمان طولانی برای اجرای آزمایش‌ها بکار می‌برد. اگرچه، ما باید اظهار کنیم که این یک نمونه کوچک از تمام امکانات سازگار سازی است که در طی توصیفات از اهدافمان در مورد این مسئله متذکر شدیم. اینجا تصمیمات بسیار کوچکتری وجود دارد که می‌توانیم به شیوه‌های دیگری دنبال کنیم (برای مثال، آستانه خرید و فروش، سیستم‌های دیگر آموزشی و غیره). این بدین معنی است که هر سری در فضای این برنامه منابع محاسباتی عظیمی نیاز دارد برای اجرای یک مدل انتخاب شده مناسب تلاش می‌کند. این به طور واضحی خارج از حوزه این کتاب است. هدف ما اینجا این است که خواننده با راهنمای روش شناختی مناسبی فراهم کنیم و سعی نکنیم بهترین سیستم تجاری برای این اطلاعا ویژه را بیابیم.

3-6-3 تجزیه و تحلیل نتایج

کد فراهم شده در بخش قبلی پنج فایل عددی با اهداف مشتمل بر نتایج تمام متغیرها، مشتمل بر 5 ، earth.Rdata، svmC.Rdata، svmR.Rdata سیستم یادگیری را تولید می‌کند. این فایل‌های اطلاعاتی نامیده می‌شود. هر کدام از آن‌ها شامل یک هدف با نام مشابه فایل بجز nnetC.Rdata و nnetR.Rdata هستند و پکیج ما شامل چندین روش است که می‌تواند برای compExp پسوند آن است. این اهداف کلاس کاوش نتایجی که ذخیره شده است مورد استفاده قرار گیرد.

به این دلیل که شما احتمالاً تجربیات را خودتان انجام نداده‌اید، شما می‌توانید فایل را در وب پیج کتاب استفاده R پیدا کنید. آن‌ها را به کامپیوترتان دانلود کنید و سپس فرمان‌های زیر را برای بارگذاری اهداف درون نمایش دهید:

```
> load("svmR.Rdata")
> load("svmC.Rdata")
> load("earth.Rdata")
> load("nnetR.Rdata")
> load("nnetC.Rdata")
```

برای هر متغیر سیستم مبادله، ما آمار چندی از عملکرد را اندازه‌گیری می‌کنیم. بعضی وابسته به عملکرد در دوره‌ای از پیش‌بینی نشانه‌های صحیح هستند، در حالیکه بقیه به عملکردهای اقتصادی وابسته هستند زمانی که این علامت‌ها در مبادله استفاده می‌شوند. تصمیم بگیرید با توجه به تجربه‌هایمان شامل یک توازن بین تمام رتبه‌ها کدامیک بهترین مدل‌ها هستند. مدل(های) انتخاب شده ممکنه بسیار وابسته باشند به اینکه بر کدام معیار ارزش بیشتری بگذاریم. علیرغم گوناگونی رتبه‌های ارزیابی ما هنوز می‌توانیم بعضی از آن‌ها را که بیشتر مرتبط است شناسایی کنیم. در بین آمارهای پیش‌بینی منفرد، دقت به طور واضحی با اهمیت‌تر از فراخوان برای این برنامه است. در نتیجه دقت باید با علامت‌های پیش‌بینی انجام شود و اینجا فعالیت‌های مبادلاتی را همچنان که آن‌ها سبب باز شدن موقعیت‌ها هستند جلو می‌برند. رتبه پایین دقت سبب سیگنال‌های غلط می‌شود، که به معنی بازگشایی موقعیت‌ها در زمان‌های غلط است. این مطمئناً به زیان‌های بیشتری هدایت‌مان می‌کند. فراخوان این هزینه بالقوه را ندارد. فراخوان قابلیت مدل‌ها را برای تسخیر شانس‌های مبادلاتی را اندازه‌گیری می‌کند. اگر این رتبه پایین باشد، این به معنی از دست دادن شانس‌ها است اما نه هزینه‌های بالا. در این متن، ما به طور در آمار هستیم که علامت‌های دقت خرید و فروش را اندازه‌گیری prec.sb خاصی علاقه‌مند به رتبه مدل‌های در آزمایش‌های ما)، و همچنین استراتژی Ret می‌کند. در اصطلاح عملکرد تجاری، بازگشت از سیستم‌ها (آمار در آزمایشات ما) مهم است. همچنین درصد سوددهی مبادله RetOverBH بازده بالاتر از خرید و نگهداری () باشد. در اصطلاح تجزیه و تحلیل ریسک، PercProf تجاری مهم است که باید به طور وضوح بالای 50% (آمار) همزمان نگاه کنیم. MaxDD این بی‌معنی است که به هر دو ارزش نسبت شارپ و ماکزیمم رسم به پایین () بکار گرفته شود. اگرچه با متغیرهای عددی داده compExp می‌تواند برای بارگذاری اهداف summary() تابع شده و عملکرد آماری آن‌ها، خروجی می‌تواند در این مورد طاقت فرسا باشد.

است که بوسیله پکیج ما فراهم می‌شود. با این تابع ما می‌تواند rankSysms() یک تناوب استفاده از تابع بالای چارت را برای ارزیابی آمارها در جایی که علاقه‌مند هستیم را با اشاره به بهترین مدل‌ها و رتبه‌های آن‌ها بدست آوریم:

```

> tgtStats <- c('prec.sb','Ret','PercProf',
+             'MaxDD','SharpeRatio')
> allSysRes <- join(subset(svmR,stats=tgtStats),
+                 subset(svmC,stats=tgtStats),
+                 subset(nnetR,stats=tgtStats),
+                 subset(nnetC,stats=tgtStats),
+                 subset(earth,stats=tgtStats),
+                 by = 'variants')
> rankSystems(allSysRes,5,maxs=c(T,T,T,F,T))

```

\$SP500

\$SP500\$prec.sb

	system	score
1	slide.svmC.v5	1
2	slide.svmC.v6	1
3	slide.svmC.v13	1
4	slide.svmC.v14	1
5	slide.svmC.v21	1

\$SP500\$Ret

	system	score
1	single.nnetR.v12	97.4240
2	single.svmR.v11	3.4960
3	slide.nnetR.v15	2.6230
4	single.svmC.v12	0.7875
5	single.svmR.v8	0.6115

\$SP500\$PercProf

	system	score
1	grow.nnetR.v5	60.4160
2	grow.nnetR.v6	60.3640
3	slide.svmR.v3	60.3615
4	grow.svmR.v3	59.8710
5	grow.nnetC.v1	59.8615

\$SP500\$MaxDD

	system	score
1	slide.svmC.v5	197.3945
2	slide.svmC.v6	197.3945

3	grow.svmC.v5	197.3945
4	grow.svmC.v6	197.3945
5	slide.svmC.v13	399.2800

\$SP500\$SharpeRatio
system score

1	slide.svmC.v5	0.02
2	slide.svmC.v6	0.02
3	slide.svmC.v13	0.02
4	slide.svmC.v14	0.02
5	slide.svmC.v21	0.02

برای انتخاب یک بخش از اطلاعات ذخیره شده در این compExps می‌تواند برای اهداف subnet() اهداف بکار رود. در این مورد ما می‌توانیم تنها یک زیرمجموعه از این آمارهای تخمینی را انتخاب کنیم. سپس قرار می‌دهیم. join() ، با استفاده از تابع compExp ما تمام متغیرهای تجاری را با هم در یک هدف واحد را در خلال ابعاد مختلفی به هم ملحق کند. در این مورد این منطقی compExp این تابع می‌تواند هدف‌های است که آن‌ها را با سیستم متغیرها به هم متصل کنیم، همچنانکه تمام شرایط آزمایشی دیگر نیز به همین را برای دستیابی به بالای 5 رتبه در میان تمام سیستم‌های rankSystems() شکل هستند. در نهایت، ما تابع مبادلاتی بری آمارهایی که انتخاب کرده‌ایم استفاده می‌کنیم. مفهوم بهترین رتبه با هر استاندارد تغییر که سایر اوقات ما کمترین بها را می‌خواهیم. می‌کند. گاهی اوقات ما بزرگ‌ترین ارزش را می‌خواهیم، در حالی که به شما اجازه می‌دهد آمارها که ماکزیمم هستند را rankSystems() این می‌تواند با ماکزیمم پارامتر از تابع مشخص کنید.

که به این پنج نتیجه بالایی نگاه می‌کنیم اولین چیزی که بدان توجه می‌کنیم این است که همه آن‌ها وقتی هستند. الگوی قابل توجه دیگر این است که تقریباً همه این متغیرها از برخی nnet یا svm شامل الگوریتم مکانیسم‌های ایجاد پنجره استفاده می‌کنند. این امر برخی شواهد از مزیت‌های این انتخاب را بیش از رویکرد مدل واحد فراهم می‌کند که می‌تواند به عنوان یک تایید تاثیرات تغییر رژیم بر روی این داده‌ها را مورد بررسی قرار دهد. ما همچنین می‌توانیم چندین رتبه عالی (و مشکوک) مشاهده کنیم، به طور مثال در مورد واژه دقت از علامت‌های خرید/فروش. بدست آمدن 100% دقت عجیب به نظر می‌رسد. یک بررسی دقیق‌تر از نتایج این سیستم‌ها آشکار خواهد کرد که این رتبه در سایه تعداد خیلی کوچکی از نشانه‌ها در خلال دوره تست 5 ساله قابل دستیابی است.

```
> summary(subset(svmC,
+               stats=c('Ret','RetOverBH','PercProf','NTrades'),
+               vars=c('slide.svmC.v5','slide.svmC.v6')))
```


== خلاصه آزمایش مونت کارلو ==

20 repetitions Monte Carlo Simulation using:

seed = 1234

train size = 2540 cases

test size = 1270 cases

* Datasets :: SP500

* Learners :: slide.svmC.v5, slide.svmC.v6

* خلاصه نتایج آزمایش:

-> Dataset: SP500

*Learner: slide.svmC.v5

	Ret	RetOverBH	PercProf	NTrades
avg	0.0250000	-77.10350	5.00000	0.0500000
std	0.1118034	33.12111	22.36068	0.2236068
min	0.0000000	-128.01000	0.00000	0.0000000
max	0.5000000	-33.77000	100.00000	1.0000000
invalid	0.0000000	0.00000	0.00000	0.0000000

*Learner: slide.svmC.v6

	Ret	RetOverBH	PercProf	NTrades
avg	0.0250000	-77.10350	5.00000	0.0500000
std	0.1118034	33.12111	22.36068	0.2236068
min	0.0000000	-128.01000	0.00000	0.0000000
max	0.5000000	-33.77000	100.00000	1.0000000
invalid	0.0000000	0.00000	0.00000	0.0000000

در نتیجه، در بیشتر این روش‌ها با یک میانگین بازده 0/25% بالاتر از دوره آزمایشی که 77/1%- زیر خرید خام و استراتژی نگهداری هست یک مبادله تکی تجاری ساخته شده است. این مدل‌ها کاملاً بلااستفاده هستند. یک تذکر نهایی بر این طبقه‌بندی جهانی این است که نتایج در اصطلاح ماکزیم ترسیم به پایین نمی‌تواند به همان بدی بررسی شود در حالیکه نسبت رتبه شارپ کاملاً ناامید کننده است. برای دستیابی به بعضی نتایج بر ارزش تمام این متغیرها، ما نیاز داریم برخی محدودیت‌ها را به برخی آمارها اضافه کنیم. اجازه دهید ارزش‌های کمینه زیر را فرض کنیم: ما می‌خواهیم 1) یک عدد منطقی از میانگین مبادلات، مثلاً بیش از 20؛

(2) یک میانگین بازده که باید حداقل بزرگتر از 0/5% باشد (کمترین رتبه عمومی داده شده به این سیستم‌ها)؛
 (3) و همچنین یک درصد از مبادلات سودده بالاتر از 40%. ما هم اکنون خواهیم دید اگر اینجا برخی سیستم‌های مبادلاتی که این محدودیت را راضی می‌کند وجود داشته باشد.

```
> fullResults <- join(svmR, svmC, earth, nnetC, nnetR, by = "variants")
> nt <- statScores(fullResults, "NTrades")[[1]]
> rt <- statScores(fullResults, "Ret")[[1]]
> pp <- statScores(fullResults, "PercProf")[[1]]
> s1 <- names(nt)[which(nt > 20)]
> s2 <- names(rt)[which(rt > 0.5)]
> s3 <- names(pp)[which(pp > 40)]
> namesBest <- intersect(intersect(s1, s2), s3)

> summary(subset(fullResults,
  stats=tgtStats,
  vars=namesBest))
```

== خلاصه‌ای از آزمایش مونت کارلو ==

20 repetitions Monte Carlo Simulation using:

seed = 1234

train size = 2540 cases

test size = 1270 cases

* Datasets :: SP500

* Learners :: single.nnetR.v12, slide.nnetR.v15, grow.nnetR.v12

* خلاصه‌ای از نتایج آزمایش

-> Dataset: SP500

*Learner: single.nnetR.v12

	prec.sb	Ret	PercProf	MaxDD	SharpeRatio
avg	0.12893147	97.4240	45.88600	1595761.4	-0.01300000
std	0.06766129	650.8639	14.04880	2205913.7	0.03798892
min	0.02580645	-160.4200	21.50000	257067.4	-0.08000000

```
max 0.28695652      2849.8500 73.08000 10142084.7
0.04000000
invalid 0.00000000      0.0000 0.00000 0.0
0.00000000
```

```
*Learner: slide.nnetR.v15
prec.sb      Ret      PercProf      MaxDD      SharpeRatio
avg 0.14028491      2.62300      54.360500 46786.28
0.01500000
std 0.05111339      4.93178      8.339434 23526.07
0.03052178
min 0.03030303      -7.03000 38.890000 18453.94 -0.04000000
max 0.22047244      9.85000      68.970000 99458.44
0.05000000
invalid 0.00000000      0.00000 0.000000 0.00
0.00000000
```

```
*Learner: grow.nnetR.v12
prec.sb      Ret      PercProf      MaxDD      SharpeRatio
avg 0.18774920      0.544500 52.66200 41998.26 0.00600000
std 0.07964205      4.334151 11.60824 28252.05 0.03408967
min 0.04411765      -10.76000 22.22000 18144.11 -
0.09000000
max 0.33076923      5.330000 72.73000 121886.17 0.05000000
invalid 0.00000000      0.00000 0.00000 0.00
0.00000000
```

که در `statScores()` برای کسب انواع نام‌های تجاری جایگزین مورد رضایت محدودیت‌ها، ما از تابع و نام آماری دریافت می‌کند و به طور `compExp` این تابع یک هدف .پکیج‌مان موجود است استفاده می‌کنیم پیش فرض، میانگین نمرات از همه سیستم‌ها در این آمار را فراهم می‌کند. نتیجه یک لیست با اجزای بسیار است همچنانکه در آزمایشات دیتابیس‌هایی وجود دارد (در مورد وضعیت ما، یک مجموعه داده واحد می‌باشد) کاربر می‌تواند یک تابع را شناسایی و آن را به سه بخش برای به دست آوردن خلاصه عددی دیگری به جای میانگین آن، تقسیم کند. با استفاده از نتایج حاصل از این تابع، نام‌های رضایت‌بخش متغیرهای هر یک از محدودیت‌ها را بدست می‌آوریم. ما در نهایت نام‌هایی از متغیرها را به دست می‌آوریم که تمام محدودیت‌ها را به راضی می‌کند که فصل مشترک بین مجموعه از ارزش‌ها بدست می‌آید. `intersect()` استفاده از تابع

همانطور که مشاهده می‌کنید، فقط سه تا از انواع 240 معامله که مقایسه شدند احساس رضایت در مورد کمترین محدودیت‌ها داشته‌اند. همه آنها بازگشت به عقب داشتند و بر اساس شبکه‌های عصبی هستند. این سه “هیچ تدبیر ایجاد پنجره‌ای را استفاده single.nnet.v12 استفاده از داده‌های آموزشی متفاوت است. روش” نمی‌کند و به میانگین بازده مؤثر 97/4% دست می‌یابد. اگرچه، در صورتیکه به نتایج این سیستم دقیق‌تر نگاه کنیم، می‌بینیم که در همان زمان در یکی از تکرارها این به بازدهی از 160/4-% دست می‌یابد. این به طور واضحی یک سیستم با نشان بی‌ثباتی از نتایج به دست آمد است، همچنانکه می‌توانیم بوسیله انحراف استاندارد از بازده (650/86%) را تایید کنیم. دو سیستم دیگر حتی به رتبه‌های مشابهی دست می‌یابند. کدهای زیر یک را اجرا می‌کند: compnalysis() تجزیه و تحلیل با اهمیت آماری از نتایج استفاده شده در تابع

```
> compAnalysis(subset(fullResults,
```

```
+      stats=tgtStats,
+      vars=namesBest))
```

== تجزیه و تحلیل‌های با اهمیت آماری از نتایج مقایسه شده ==

Baseline Learner:: single.nnetR.v12 (Learn.1)

** Evaluation Metric:: prec.sb

- Dataset: SP500

	Learn.1	Learn.2 sig.2	Learn.3 sig.3
AVG	0.12893147	0.14028491	0.18774920 +
STD	0.06766129	0.05111339	0.07964205

** Evaluation Metric:: Ret

- Dataset: SP500

	Learn.1	Learn.2 sig.2	Learn.3 sig.3
AVG	97.4240	2.62300 -	0.544500 -
STD	650.8639 4.93178	4.334151	

** Evaluation Metric:: PercProf

- Dataset: SP500

	Learn.1	Learn.2 sig.2	Learn.3 sig.3
AVG	45.88600	54.360500 +	52.66200
STD	14.04880	8.339434	11.60824

** Evaluation Metric:: MaxDD

- Dataset: SP500

	Learn.1	Learn.2 sig.2	Learn.3 sig.3
AVG	1595761	46786.28 --	41998.26 --
STD	2205914	23526.07	28252.05

** Evaluation Metric:: SharpeRatio

- Dataset: SP500

154 *Data Mining with R: Learning with Case Studies*

	Learn.1	Learn.2 sig.2	Learn.3 sig.3
AVG	-0.01300000	0.01500000 +	0.00600000
STD	0.03798892	0.03052178	0.03408967

Legends:

Learners -> Learn.1 = single.nnetR.v12 ; Learn.2 = slide.nnetR.v15 ;

Learn.3 = grow.nnetR.v12 ;

Signif. Codes -> 0 '++' or '--' 0.001 '+' or '-' 0.05 ' ' 1

توجه داشته باشید که کد بالا می تواند هشدارهای ناشی از این واقعیت که برخی از سیستم ها امتیاز معتبری در برخی از آمارها را به دست نمی آورند، ایجاد کند (به عنوان مثال ، علامت های خرید یا فروش به رتبه های دقت معتبری هدایت نمی کند).

به ما می گوید که متوسط بازده Wilcoxon با وجود تنوع نتایج، آزمون با اهمیت "بیشتر از آن سیستم های دیگر با اطمینان 95% است. با این حال، با احترام به آمارهای single.nnetR.v12 دیگر این تغییرات به شکل آشکاری غلط است.

ما ممکن است یک ایده بهتر از توزیع امتیاز در برخی از این آمارها در طول 20 بار تکرارشان بوسیله ترسیم داشته باشیم: compExp هدف

```
> plot(subset(fullResults,
+ stats=c('Ret','PercProf','MaxDD'),
+ vars=namesBest))
```

نتایج حاصل از این کد در نمودار 3-8 نشان داده شده است.

امتیازهای دو سیستم استفاده شده در تدبیر ایجاد پنجره بسیار شبیه هستند به همین خاطر تشخیص آن " به وضوح متمایز هستند. ما `single.nnetR.v12` را در بین آن‌ها مشکل می‌سازد. در مقابل، نتایج حاصل از " می‌توانیم مشاهده کنیم که میانگین بازده بالا در نتیجه غیرعادی بودن آشکار (در حدود 2800%) بازده در یکی از تکرارهای آزمایش مونت کارلو، قابل دستیابی است. باقی‌مانده امتیاز برای این سیستم به نظر می‌رسد به طور آشکاری در سطح پایین‌تر امتیاز دو تای دیگر است. فقط از کنجکاو می‌توانیم به پیکربندی این سیستم بررسی کنیم: `getVariant()` تجاری خاص را با استفاده از تابع

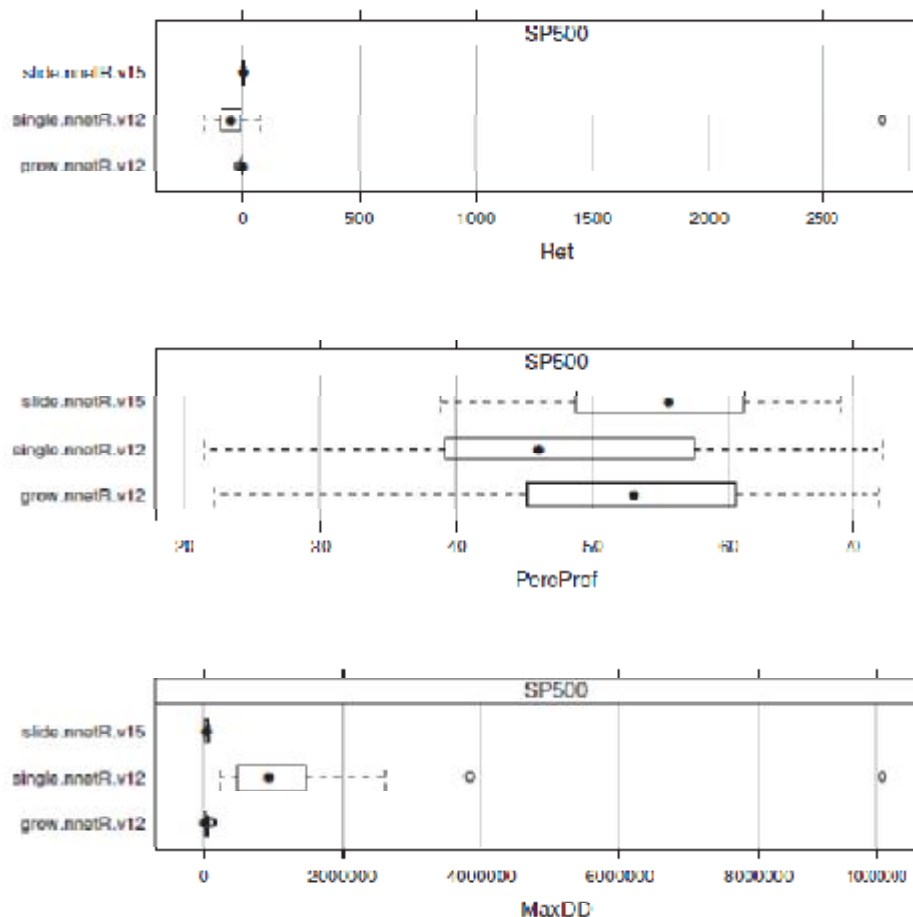
```
> getVariant("single.nnetR.v12", nnetR)
```

Learner:: "single"

Parameter values

```
learner = "nnetR"  
linout = TRUE  
trace = FALSE  
maxit = 750  
size = 10  
decay = 0.01  
policy.func = "pol3"
```

استفاده می‌کند و یک شبکه عصبی با `pol3` همانطور که می‌توانید مشاهده کنید، این از سیاست مبادلاتی ده واحد مخفی با یک نرخ تنزل 0/01 را آموزش می‌دهد.



نمودار 3-8: رتبه بهترین مبادلات در 20 تکرار

به طور خلاصه، با توجه نتایج مشخص شده، اگر ما هر کدام از تناوبات بررسی شده را انتخاب کنیم، با خواهیم پرید. با این وجود، در قسمت بعدی ما سه تا از `single.nnetR.v12` بی‌ثباتی داده شده احتمالاً از بهترین سیستم‌های تجاری‌مان را بر روی اطلاعات 9 سال آخر بکار خواهیم برد که برای ارزیابی نهایی بهترین سیستم‌ها رها می‌شود.

3-7- سیستم بازرگانی

این بخش نتایج به دست آمده توسط "بهترین" مدل در ارزیابی نهایی را ارائه می‌کند، که از مقایسه و انتخاب مراحل صرف‌نظر شده است. این دوره با قیمت جاری روزانه 9 سال شکل گرفته است و ما پنج سیستم

انتخاب شده را برای مبادله در خلال این دوره با استفاده از شبیه‌سازی مان بکار می‌بریم.

3-7-1 ارزیابی آزمون نهایی داده‌ها

برای اعمال هر یک از سیستم‌های انتخاب شده به مدت دوره ارزیابی، ما حداقل به 10 سال قبل از این دوره ارزیابی نیاز داریم. مدل‌ها با این 10 سال اطلاعات بدست خواهند آمد و سپس برای علامت پیش‌بینی برای 9 سال دوره ارزیابی‌شان درخواست خواهد شد. این پیش‌بینی‌ها ممکن است در حقیقت شامل بدست آوردن بیشتر مدل‌ها در موارد سیستم با استفاده از تدبیر ایجاد پنجره باشد. کد زیرین شامل ارزیابی آمارها از این سیستم‌ها روی دوره آزمایشی 9 ساله است:

```
> data <- tail(Tdata.train, 2540)
> results <- list()
> for (name in namesBest) {
+   sys <- getVariant(name, fullResults)
+   results[[name]] <- runLearner(sys, Tform, data, Tdata.eval)
+ }
> results <- t(as.data.frame(results))
```

ما سه تا از بهترین مدل‌ها را برای بدست آوردن پیش‌بینی‌هایشان بوسیله فراخوان آن‌ها با اطلاعات آموزشی اولیه (10 ساله) و با ارزیابی دوره همچون اطلاعات آزمایشی در یک دوره زمانی تکرار می‌کنیم. این فراخوان که در گذشته تعریف کردیم می‌باشد. نتیجه این توابع یک `grow()` و `slide()`، `single()` شامل استفاده از تابع `eval.stats()` که قبلاً دیدیم، است. در پایان این حلقه ما `eval.stats()` گروه از استانداردهای ارزیابی تولید شده توسط تابع لیست بدست آمده نتایج را به فرمت مناسب‌تری مانند جدول انتقال می‌دهیم. اجازه دهید ارزش برخی از آمارهای اصلی را بررسی کنیم:

```
> results[, c("Ret", "RetOverBH", "MaxDD", "SharpeRatio", "NTrades",
+ "PercProf")]
```

	Ret	RetOverBH	MaxDD	SharpeRatio	
NTrades					PercProf
single.nnetR.v12	-91.13	-61.26	1256121.55	-0.03	
759					44.66
slide.nnetR.v15	-6.16	23.71	107188.96	-0.01	132 48.48
grow.nnetR.v12	1.47	31.34	84881.25	0.00	89 53.93

داشته سال به نتایج مثبت دستیابی همانطور که تایید می‌کنید، تنها یکی از سه سیستم تجاری مدت 9 با یک رتبه بازده پایین‌تر از 91/13%-single.nnetR.v12 است. باقی سیستم‌ها، با تایید بی‌ثباتی سیستم به وضوح بهتر است و "grow.nnetR.v12" پول از دست داده‌اند. در میان دوتای دیگر، به نظر می‌رسد روش با این نه تنها با بازده مثبت بلکه همچنین با ترسیم رو به پایین کوچک‌تر و درصد سوددهی تجاری بالای 50%. حال این دو سیستم به وضوح بالاتر از بازار در دوره تست این کار را با بازده بیش از خرید و نگهداری 23/7% و 31/4% است.

بهترین مدل دارای مشخصات زیر است:

```
> getVariant("grow.nnetR.v12", fullResults)
```

Learner:: "grow"

Parameter values

```
learner = "nnetR"
relearn.step = 120
inout = TRUE
trace = FALSE
maxit = 750
size = 10
decay = 0.001
policy.func = "pol2"
```

ما هم‌اکنون با تجزیه و تحلیل عمیق‌تری از عملکرد بهترین سیستم تجاری برای ادامه در سراسر دوره را ارزیابی می‌کنیم. برای اینکه این کار ممکن باشد، لازم است ثبت تجاری سیستم در خلال این دوره را بدست این هدف را باز نمی‌گرداند بنابراین ما نیاز داریم این را به شکل دیگری به grow() آوریم. تابع دست آوریم:

```
> model <- learner("MC.nnetR", list(maxit = 750, linout = T,
+   trace = F, size = 10, decay = 0.001))
> preds <- growingWindowTest(model, Tform, data, Tdata.eval,
+   relearn.step = 120)
> signals <- factor(preds, levels = 1:3, labels = c("s", "h",
+   "b"))
> date <- rownames(Tdata.eval)[1]
> market <- GSPC[paste(date, "/", sep = "")][1:length(signals),
```

```
+ ]
> trade.res <- trading.simulator(market, signals, policy.func = "pol2")
```

نمودار 9_3 ثبت تجاری این سیستم را رسم می‌کند و به شرح زیر است:

```
> plot(trade.res, market, theme = "white", name = "SP500 - final test")
```

تجزیه و تحلیل نمودار 9_3 نشان می‌دهد که سیستم از طریق یک دوره طولانی، یعنی از اواسط 2003 تا اواسط سال 2007 تقریباً هیچ فعالیت تجاری نداشته است. این کاملاً شگفت‌انگیز است زیرا یک دوره افزایش قابل توجهی در سوددهی بازار وجود داشت. این به نوعی نشان دهنده اینست که سیستم به خوبی که می‌توانست عمل نکرده بود، با این وجود نتایج جهانی مورد توجه قرار دارد. همچنین قابل توجه است که سیستم به طور فوق‌العاده خوبی در طول گرایش رو به پایین سال‌های 2000 تا 2003 و همچنین در دوره بحران مالی سالهای 2007 تا 2009 زنده ماند.



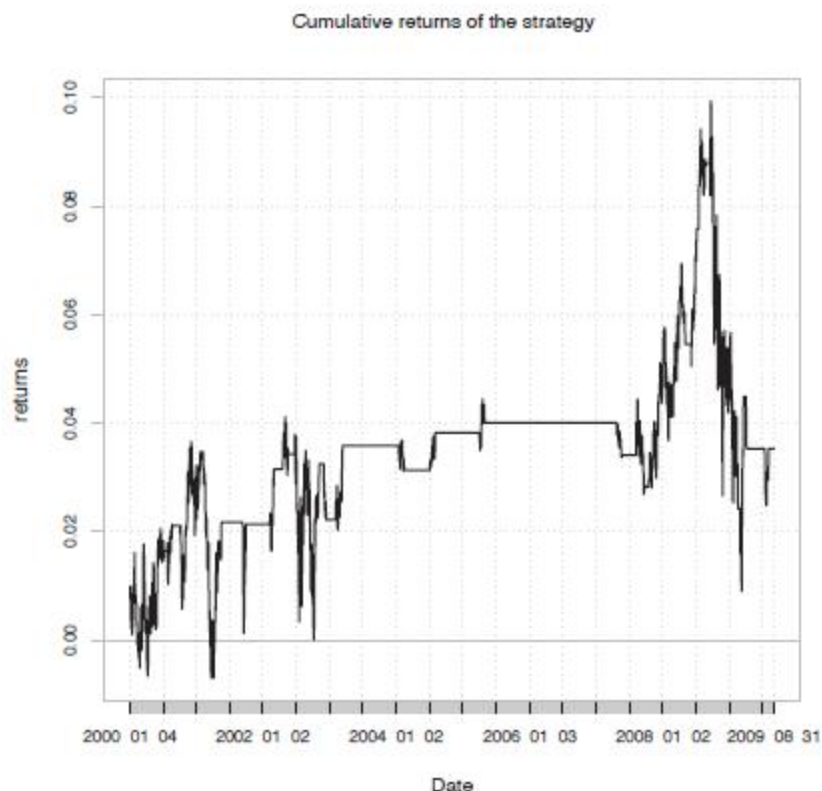
grow.nnetR.v12 شکل 9_3: نتایج دوره ارزیابی مالی از سیستم

پکیج تجزیه و تحلیل عملکرد یک گروه پرقدرت از ابزارها برای تجزیه و تحلیل عملکرد هر سیستم تجاری فراهم می‌کند. اینجا ما یک نگاه اجمالی به برخی از این ابزارها برای بدست آوردن دیدگاه بهتری به عملکرد سیستم تجاری مان می‌پردازیم. ابزارهای این پکیج بر روی بازده استراتژی در دست ارزیابی کار می‌کند. بازگشت استراتژی ما می‌تواند شامل موارد زیر باشد:

```
> library(PerformanceAnalytics)
> rets <- Return.calculate(trade.res@trading$Equity)
```

درصد بازدهی که ما تا کنون استفاده می‌کردیم Return.calculate() لطفا توجه داشته باشید که تابع را محاسبه نمی‌کند، هنوز این بازده با فاکتوری از 100 هم‌ارز است. شکل 10_3 بازده انباشته از استراتژی کلی تمام دوره‌های آزمایشی را نشان می‌دهد. برای بدست آوردن چنین نموداری، این معنی‌دار است که کد ذیل را اجرا کنیم:

```
> chart.CumReturns(rets, main = "Cumulative returns of the strategy",
+ ylab = "returns")
```



grow.nnetR.v12 شکل 10_3 : بازده انباشته دوره ارزیابی پایانی از سیستم

برای بیشتر دوره‌ها، این سیستم در وضعیت مثبت خود قرار دارد، که در نیمه سال 2008 به اوج خود در قله با رقم بازده 10% رسیده است.

این یک استفاده مفید مکرر برای دستیابی به اطلاعات راجع به بازگشت در طی یک سال یا حتی مبنای ماهانه است. پکیج تجزیه و تحلیل عملکرد برخی ابزارها را برای کمک به این نوع از تجزیه و تحلیل فراهم می‌کند، برای مثال تابع `yearlyReturn()` :

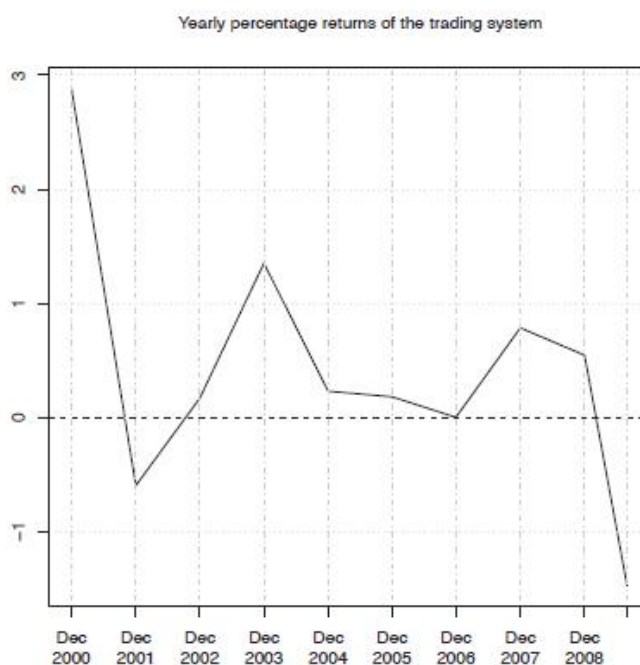
```
> yearlyReturn(trade.res@trading$Equity)
```

```
yearly.returns
2000-12-29 0.028890251
2001-12-31 -0.005992597
2002-12-31 0.001692791
2003-12-31 0.013515207
2004-12-31 0.002289826
2005-12-30 0.001798355
2006-12-29 0.000000000
2007-12-31 0.007843569
2008-12-31 0.005444369
```

2009-08-31 -0.014785914

نمودار 11-3 این اطلاعات گرافیکی را ارائه می دهد و ما می توانیم مشاهده کنیم که اینجا تنها 2 سال با بازده منفی وجود دارد.

```
> plot(100*yearlyReturn(trade.res@trading$Equity),  
+      main='Yearly percentage returns of the trading system')  
> abline(h=0,lty=2)
```



grow.nnetR.v12 نمودار 11-3: درصد سالانه بازگشت از سیستم

حتی جزئیات بیشتری از اطلاعات را با جدولی از یک استراتژی درصد `table.calendarReturns()` بازده ماهانه (ستون آخر مجموع باقیمانده در طول سال) فراهم می کند:

```
> table.CalendarReturns(rets)
```

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
Nov	Dec	Equity								
2000	-0.5	0.3	0.1	0.2	0	0.2	0.2	0.0	0.0	0.4
	-0.2	1.0								0.4

2001	0.0	-0.3	0.2	-0.1	0	0.0	0.0	0.0	0.4	0.0	0.0
	0.0	0.3									
2002	0.0	-0.1	0.0	-0.2	0	0.0	0.2	0.0	-0.3	-0.1	0.0
	0.0	-0.5									
2003	0.0	-0.1	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	-0.1									
2004	0.1	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0									
2005	0.0	0.0	0.0	-0.2	0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	-0.2									
2006	0.0	0.0	0.0	0.0	0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	NA									
2007		0.0	0.0	0.0	0.2	0	0.0	0.0	-0.2	0.0	-0.2
	0.2	0.1	0.0								
2008	-0.3	0.5	0.1	0.1	0	0.0	0.3	0.0	0.9	0.3	0.2
	0.3	2.3									
2009	-0.5	0.0	-0.2	0.0	0	0.0	0.0	0.0	NA	NA	NA
	NA	-0.6									

این جدول به وضوح نشان می‌دهد در دوره طولانی مدتی که سیستم غیرفعال می‌ماند بازده با صفرهای بسیاری همراه است.

در نهایت، ما یک مثال از برخی ابزارها بوسیله پکیج تجزیه و تحلیل عملکرد برای بدست آوردن اطلاعات در `table.DownsideRisk()` را ارائه می‌کنیم. تجزیه و تحلیل از استراتژی استفاده شده در تابع

`> table.DownsideRisk(rets)`

	Equity
Semi Deviation	0.0017
Gain Deviation	0.0022
Loss Deviation	0.0024
Downside Deviation (MAR=210%)	0.0086
Downside Deviation (Rf=0%)	0.0034
Downside Deviation (0%)	0.0034
Maximum Drawdown	-0.0822
Historical VaR (95%)	-0.0036
Historical ES (95%)	-0.0056
Modified VaR (95%)	-0.0032

این تابع اطلاعاتی در رابطه با چندین ریسک اندازه‌گیری می‌دهد از جمله اینکه ما حداکثر درصد رسم به پایین را پیدا می‌کنیم و همچنین نیمه انحرافات که اخیراً به عنوان ریسک بهتر اندازه‌گیری نسبت شارپ با تکرار بیشتر پذیرفته می‌شود. اطلاعات بیشتری روی این آمار می‌تواند در صفحه کمک‌های پکیج تجزیه و تحلیل عملکرد یافت شود.

grow.nnetR.v12 رویهم رفته، تجزیه و تحلیلی که ما انجام دادیم نشان می‌دهد که سیستم مبادلاتی یک بازده کوچکی با ریسک بزرگی از دوره آزمایشی 9 ساله فراهم می‌کند. علارغم اینکه وضوح بالای خرید خام و استراتژی نگهداری، این سیستم آماده مدیریت پولهایتان نیست! همچنان، باید بگوییم که این مورد انتظار بود. این یک مشکل کاملاً اساسی با احتمالات و متغیرهای بسیار است که برخی از آن‌ها را ما در این فصل شرح می‌دهیم. این بسیار شگفت‌آور خواهد بود اگر گروه‌های کوچکی از احتمالاتی که می‌آزماییم ما را به سیستم مبادلاتی موفق‌تری رهنمون سازد. البته این هدف این مورد پژوهی ما نبوده است. هدف ما فراهم کردن خواننده برای فرایندهایی است که روش شناختی خواننده می‌شود و نه اینکه در عمق مسئله برای بهترین سیستم تجاری جستجو کنیم که از این روش شناختی استفاده می‌کند. هدف ما آشنا ساختن خواننده با استفاده از روش‌هایی به انجام جستجوی عمیق‌تر برای بهترین سیستم تجاری بود.

3-7-2 یک سیستم تجاری آنلاین

اجازه دهید فرض کنیم خوشحالیم که با سیستم تجاری ما نیز بهبود و توسعه داشته‌ایم. چگونه ما می‌توانیم از تجارت بلادرنگ در بازار استفاده کنیم؟ در این بخش ما طرح مختصری از یک سیستم با کاربردهای آن را ارائه می‌دهیم. سیستم خودکار به ما پیشنهادی به شرح زیر می‌دهد. در پایان هر روز سیستم به شکل خودکار فراخوان خواهد شد. این سیستم باید (1) شامل هر کدام از داده‌های جدید قابل دسترسی باشد. (2) هر گونه مدل سازی و اقداماتی که ممکن است نیاز باشد انجام داده شود و (3) مجموعه‌ای از دستورات را به عنوان که به آن نیاز داریم تولید کند. خروجی از فراخوان

"trader.R اجازه دهید فرض کنیم توسعه سیستم ما به صورت کد ذخیره شده بر روی یک فایل به نام " است. راه حل این است که این برنامه را در پایان هر روز با توجه به سیستم عملیاتی که استفاده می‌کنید " وجود دارد که ما crontab فراخوان نماییم. در سیستم‌های مبتنی بر یونیکس معمولاً یک جدول به نام "به می‌توانیم نوشته‌های اضافه با برنامه‌هایی که باید به طور منظم توسط سیستم عامل اجرا شود را استفاده کنیم. ویرایش این جدول می‌تواند در خط فرمان بوسیله این دستور انجام گیرد:

```
shell> crontab -e
```

نحو ورودی در این جدول منطقی و ساده است و بوسیله مجموعه‌ای از پرونده‌ها که به تناوب توضیح داده trader.R شکل می‌گیرد و در نهایت فرمان اجرا می‌شود. در زیر شما می‌توانی مثالی بیابید که باید برنامه را هر هفته ساعت 19:00 اجرا نماید:

```
0 19 * * 1-5 /usr/bin/R --vanilla --quiet < /home/xpto/trader.R
```

دو ورودی اولیه نشان دهنده دقیقه و ساعت است. به ترتیب سوم روز و چهارم ماه و ستاره بدان معنی است که این برنامه باید برای همه نمونه‌هایی که در این زمینه‌ها اجرا می‌شود به کار رود. ورودی پنجم روز هفته با نمایندگی روز دوشنبه است و برای تعیین فواصل از جدا کننده "-" استفاده شده است. در نهایت، ما برنامه را با کد منبع بازرگانان است.R اجرا می‌کنیم که در این مورد یک فراخوان به اجرا می‌گردد که به شکل زیر است: trader.R الگوریتم کلی در برنامه

- خواندن وضعیت کنونی بازرگان
- گرفتن تمام اطلاعات در دسترس جدید
- چک کردن اینکه آیا نیاز به بازآموزی مدل وجود دارد
- بدست آوردن یک سیگنال پیش‌بینی برای امروز
- با این سیگنال، فراخوانی تابع سیاست برای بدست آوردن دستورات
- خروجی دستورات امروز

وضعیت جاری بازرگان باید یک ساختار داده تنظیم کند که اطلاعاتی که مورد نیاز است از طریق NNET، فعالیتهای روزانه بازرگان خلاصه‌سازی شود ذخیره گردد. در مدل ما این باشد شامل مدل اخیر پارامترهای یادگیری استفاده شده در آن، اطلاعات آموزشی بدست آمده در مدل، ویژگی مدل اطلاعات وابسته، سن مدل (مهم است بدانید کی بازآموزی شده است)، اطلاعاتی برطبق سیستم ثبت مبادله تا امروز و این موقعیت بازگشایی جاری باشد. در حالت ایده‌آل، این اطلاعات باید در دیتابیس باشد و بازرگان می‌تواند از طریق با این سیستم‌ها به آن دسترسی یابد (بخش 2-3 را ببینید). لطفا توجه داشته باشید که اطلاعات بر R واسط روی موقعیت باز باید خارج از سیستم به روز رسانی شود همچنانکه این بازاری است که زمان برای بازگشایی و بستن موقعیت‌ها را مشخص می‌کند، برخلاف شبیه‌ساز ما جایی که ما فرض می‌کنیم که همه سفارشاتمان تکمیل شده است در آغاز روز می‌باشد.

به دست آوردن اطلاعات جدید در دسترس آسان است، اگر ما خصوصیات مدل را داشته باشیم. همچنانکه می‌تواند برای نو کردن بسته اطلاعاتی ما با بیشترین getModelData() در بخش 2-3 اشاره شد، تابع قیمت‌های به روز شده استفاده گردد.

بالا رود که باید به حافظه سپرده شود در relearn.step لازم است مدل بازآموزی شود اگر سن پارامتر را فراخوانی کنیم تا مدل MC.nntR() پیوستگی با تمام پارامترهای مدل. اگر مسئله این است، سپس باید تابع

جدیدی با پنجره جدید از اطلاعات بازگشایی کنیم. همانطور که بهترین بازرگانان استراتژی پنجره‌های رو به رشد را استفاده می‌کنند، بسته آموزشی به طور ثابتی رشد خواهد کرد، که ممکنه شروع کند به مشکل ایجاد کردن اگر خیلی بزرگ شود و نتواند با حافظه کامپیوتر تطبیق یابد. اگر آن مسئله اتفاق افتاد، ما می‌توانیم به بررسی کردن فراموشی اطلاعات خیلی قدیمی پردازیم، به موجب آن بسته آموزشی به سبب پذیرفتنی هرس می‌شود (کوچک می‌گردد).

با `predict()` در نهایت ما باید علامت‌ها را امروز پیش‌بینی کنیم. این بدین معنی است که فراخوان تابع یک مدل جاری برای بدست آوردن یک پیش‌بینی برای آخرین سطر بسته آموزشی متعلق به امروز است. با داشتن این پیش‌بینی، ما می‌توانیم تابع خط‌مشی تجاری را با پارامترهای مناسب برای بدست آوردن مجموعه‌ای از سفارشات خروجی برای امروز فراخوانی کنیم. این باید نتیجه نهایی برنامه امروز باشد. این طرح خلاصه باید شما را با اطلاعات کافی برای اجرای چنین سیستم‌های تجاری آنلاینی آماده کند.

3-8- خلاصه

هدف اصلی از این فصل معرفی یک برنامه واقعی از داده کاوی به خواننده است. برنامه ملموس که شرح داده شد شامل چندین چالش جدید است، برای مثال (1) مدیریت کردن اطلاعات سری زمانی (2) خرید و فروش بایک سیستم بسیار پویا با تغییرات احتمالی رژیم و (3) حرکت از مدل پیش‌بینی به فعالیت‌های واقعی در فضای برنامه.

در اصطلاح روش شناختی به شما چند موضوع جدید معرفی می‌کنیم:

- مدل‌سازی سری زمانی
- مدیریت کردن تغییرات رژیم با مکانیسم ایجاد پنجره
- شبکه‌های عصبی مصنوعی
- مکانیسم بردار حمایتی
- حداقل رگرسیون چند متغیره تطبیقی
- ارزیابی مدل‌های سری زمانی با روش مونت کارلو
- چند ارزیابی آماری جدید مربوط به پیش‌بینی‌هایی که اتفاقات نادری هستند یا با عملکرد تجاری مالی هستند

ما شرح می‌دهیم از منظر یادگیری

- چگونه اطلاعات سری زمانی را مدیریت کنیم
- چگونه اطلاعات را از منابع مختلف مانند مبنای اطلاعات بخوانیم
- چگونه چندین مدل جدید بدست آوریم (SVMs, ANNs and MARS)
- چگونه چندین پکیج ویژه به مدل مالی اختصاص دهیم

فصل 4 مترجمان:

مهدی سالک

دنیا نوری

کیوان ثابتی

شناسایی فعالیت های جعلی

سومین بررسی موردی معرفی مشکل کلی شناسایی مشاهدات غیر معمول یک پدیده و یافتن بررسی های مختلف و کمیاب است . کاربرد موثر باید با اقدامات مجموعه محصولاتی انجام شود که بوسیله فروشندگان یک شرکت گزارش شده است . هدف از این اقدام یافتن گزارشات غیر متجانس فعالیت ها می باشد که ممکن است نشان دهنده تلاش های جعلی توسط برخی فروشندگان باشد . پیامد فرایند استخراج اطلاعات از فعالیت های بازرسی پیشین شرکت حمایت می نماید . با ارائه مقدار محدود منابع که برای این فعالیت بازرسی اختصاص داده شده است ما قصد داریم نوعی رتبه بندی محتمل بر قلب را به عنوان پیامد فرایند فراهم نماییم . این رتبه بندی ها این امکان را به شرکت می دهند تا از طریق روش های سودمند منابع خود را بازرسی کنند . چنین فعالیت هایی در حوزه های مختلف و به طور متناوب قابل انجام است همچون اقدامات کارت اعتباری ، بازبینی بیانیه های مالیاتی و در این فصل ما فعالیت های مختلف و جدید استخراج اطلاعات را بیان می نماییم که عمدتاً شامل موارد زیر است : 1- شناسایی غیر متعارف 2- دسته بندی 3- مدل های پیش بینی نیمه نظارتی شرح موضوع و اهداف :

شناسایی تقلب و فریبکاری در واقع بخش مهمی برای کاربرد بالقوه تکنیک های استخراج اطلاعات می باشد که نتایج اقتصادی و اجتماعی دارد و معمولاً با فعالیت های غیر قانونی همراه است . بر اساس دیدگاه تحلیل اطلاعات ، فعالیت های متقلبانه معمولاً با مشاهدات غیر معمول همراه است و فعالیت هایی هستند که از حالت نرمال و هنجار دور شده و منحرف گشته اند . چنین انحرافات که از مسیر طبیعی خارج گشته اند به صورت کلی امور غیر متعارفی در اصول مختلف تحلیل اطلاعات محسوب می شوند . در حقیقت ، تعریف استاندارد از

امور غیر متعارف این است: مشاهداتی که نسبت به دیگر موارد مشابه از مسیر نرمال منحرف گشته اند و موجب برانگیختگی بدگمانی ها می گردد و بوسیله مکانیسم مختلفی ارتقا می یابد.

اطلاعاتی که در این تحقیق موردی استفاده می نماییم به فعالیت هایی مربوط می شود که توسط فروشندگان یک شرکت ارائه شده است. این فروشندگان مجموعه ای از محصولات شرکت را فروخته و فروش را بر اساس دوره ای مشخص گزارش نموده اند. اطلاعات حاصله و در دسترس با گزارشات دوره زمانی کوتاه مرتبط است. فروشندگان آزاد هستند تا قیمت فروش را بر اساس خط مشی ها و بازار تنظیم کنند. در انتهای هر ماه، آنها به شرکت بر می گردند تا معاملات خود را گزارش کنند. هدف از کاربرد استخراج اطلاعات، کمک به اصلاح راستگویی در گزارشات مربوط به تجارب گذشته شرکت است که خطاها و تلاش های فریبکارانه در گزارشات را شناسایی می نماید. چنین اقدامی فراهم کننده نوعی رتبه بندی در میان گزارشات است که بر اساس میزان احتمال تقلب کاری می باشد. همچنین رتبه بندی این امکان را فراهم می کند تا منابع بازرسی محدود شرکت به گزارشات اختصاص یابد و سیگنال های سیستم بیشتر مشکوک شوند.

اطلاعات در دسترس:

اطلاعاتی که ما در دسترس داریم در واقع منبع افشا نشده بوده و قبلاً بررسی شده است. هر کدام از ردیف های 401, 146 جدول اطلاعات شامل اطلاعات موجود در گزارش می باشد که توسط فروشندگان ارائه شده است. این اطلاعات شامل ID فردی، ID محصول، کمیت و ارزش کل گزارش شده است که توسط فروشنده گزارش شده است. این اطلاعات قبلاً در شرکت مورد تحلیل قرار گرفته است. نتیجه این تحلیل در ستون آخر مشخص شده است که نتیجه بازرسی برخی معاملات توسط شرکت است. به طور خلاصه، مجموعه اطلاعات که مورد استفاده قرار گرفته است در ستون های زیر ارائه شده است:

- ID: یک فاکتور با ID فروشنده
- Prod: یک فاکتور که نشان دهنده ID محصول فروخته شده است.
- Quant: تعداد واحدهای فروخته شده گزارش شده
- Val: مقدار ارزش مالی کل فروش
- Insp: یک فاکتور با سه مقدار ارزشی ممکن:
- OK: اگر معامله مورد بازبینی قرار گرفته باشد و صحت آن توسط شرکت مورد بررسی قرار گرفته باشد.
- Fraud: اگر مشخص گردد که معامله به صورت فریبکارانه بوده است.
- Unkn: اگر معامله توسط شرکت مورد بازبینی و بررسی قرار نگرفته باشد.

بارگذاری اطلاعات در R:

مجموعه اطلاعات ما در بسته کتاب و یا وب سایت کتاب موجود است. در وب سایت کتاب نیز فایل Rdata موجود است که شامل چهارچوب اطلاعات با مجموعه اطلاعات می باشد. برای استفاده از این فایل شما باید آنرا در دایرکتوری کامپیوتر خود دانلود نمایید و سپس فرمان را ثبت نمایید.

```
> load("sales.Rdata")
```

اکنون اطلاعات در دایرکتوری که شما فایل را دانلود کرده اید وارد شده است . این بارگذاری از چهارچوب اطلاعاتی از فایل sales گفته می شود . اگر تصمیم بگیرید از فایل های بسته بندی کتاب استفاده نمایید شما

```
> library(DMwR)
```

```
> data(sales)
```

باید از دستورات زیر استفاده نمایید .

مجدداً در این بخش چهارچوب اطلاعات بدست آمده sales گفته می شود که در زیر ردیف های آن نشان داده شده است .

```
> head(sales)
```

	ID	Prod	Quant	Val	Insp
1	v1	p1	182	1665	unkn
2	v2	p1	3072	8780	unkn
3	v3	p1	20393	76990	unkn
4	v4	p1	112	1100	unkn
5	v3	p1	6164	20260	unkn
6	v5	p2	104	1155	unkn

کاوش مجموعه اطلاعات :

برای کسب مرور داخلی از ویژگیهای آماری اطلاعات ما از خلاصه عملکرد زیر استفاده می نمایم .

```
> summary(sales)
```

ID		Prod		Quant		Val	
v431	: 10159	p1125	: 3923	Min.	: 100	Min.	: 1005
v54	: 6017	p3774	: 1824	1st Qu.:	: 107	1st Qu.:	: 1345
v426	: 3902	p1437	: 1720	Median	: 168	Median	: 2675
v1679	: 3016	p1917	: 1702	Mean	: 8442	Mean	: 14617
v1085	: 3001	p4089	: 1598	3rd Qu.:	: 738	3rd Qu.:	: 8680
v1183	: 2642	p2742	: 1519	Max.	: 473883883	Max.	: 4642955
(Other):	372409	(Other):	388860	NA's	: 13842	NA's	: 1182
Insp							
ok	: 14462						
unkn	: 385414						
fraud:	1270						

همچنین تعداد قابل توجهی محصولات و فروشندگان وجود دارند و ما می توانیم استفاده از عملکرد nlevels را تأیید نمایم .

```
> nlevels(sales$ID)
```

```
[1] 6016
```

```
> nlevels(sales$Prod)
```

```
[1] 4548
```

نتیجه عملکرد نشان دهنده چندین حقیقت مرتبط در این اطلاعات است . ابتدا مقادیر قابل توجه نامشخصی در ستون های Quant و Val وجود دارد . اگر هر دو در همان زمان رخ دهند مطمئناً مشکل ساز خواهد بود به طوریکه این مسئله نشان دهنده گزارش معامله بدون اطلاعات مهم در کمیت های مربوط به فروش است . اگر چنین وضعیتی وجود داشته باشد ما به راحتی می توانیم این موضوع را بررسی نماییم.

```
> length(which(is.na(sales$Quant) & is.na(sales$Val)))
```

[1] 888

همانطور که مشاهده می کنید تعداد قابل قبول معاملات مشخص شدند . با ارائه مقدار کلی معاملات ، این پرسش مطرح می گردد که آیا بهتر نیست به سادگی این گزارشات پاک گردند . ما به این موضوع در بخش های بعدی می پردازیم .

همانطور که اشاره گردید (به ویژه برای مجموعه اطلاعات بسیار بزرگ) ، شکل های مناسبی از کسب اینگونه اطلاعات وجود دارد . گرچه کد قبلی با استفاده از length() و which() ممکن است بیشتر قابل فهم باشد اما می توانیم امتیاز کسب مقادیر منطقی را بدست آوریم که در فرمان زیر آمده است.

```
> sum(is.na(sales$Quant) & is.na(sales$Val))
```

[1] 888

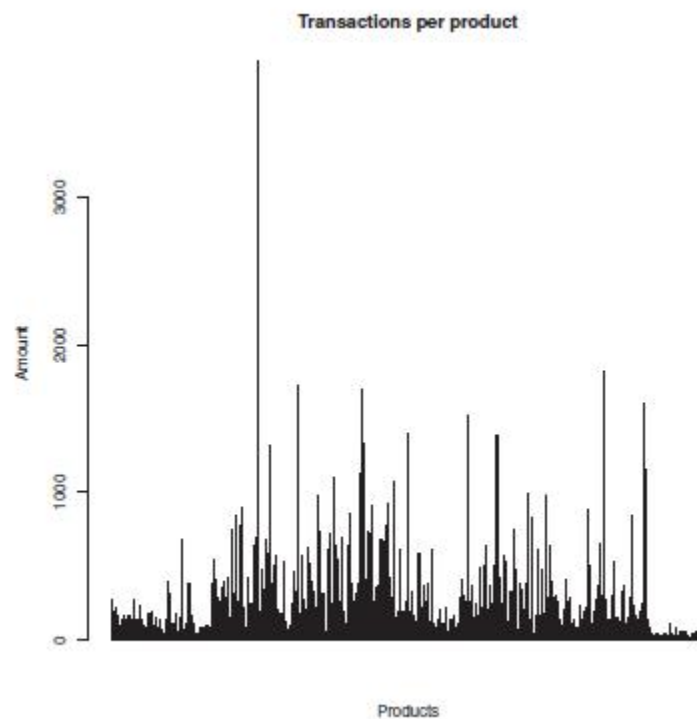
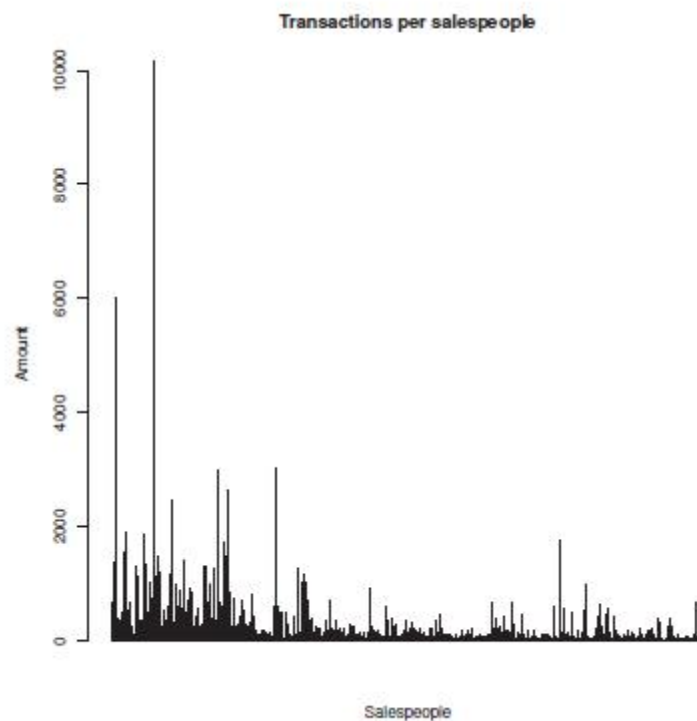
از دیگر مشاهدات قابل توجه که از عملکرد summary() بدست آمده است می توان به توزیع مقادیر و ارزش ها در ستون بازبینی اشاره نمود . بر اساس ، سهم تقلب در این بخش پایین است حتی اگر تنها گزارشات بازرسی شده را به حساب آوریم و سهم کوچکی از آن به صورت زیر است .

```
> table(sales$Insp)/nrow(sales) * 100
```

ok	unkn	fraud
3.6051712	96.0782359	0.3165930

شکل 4-1 نشان دهنده تعداد گزارشات هر فروشنده است . همانطور که شما می توانید تأیید نمایید ، تعداد نمون های فروشنده ها گوناگون است . شکل 4-2 نشان دهنده همان تعداد بر حسب محصول است . مجدداً متغیر را مشاهده می کنیم . هر دو نمونه با کد زیر بدست می آیند .

```
> totS <- table(sales$ID)
> totP <- table(sales$Prod)
> barplot(totS, main = "Transactions per salespeople", names.arg = "",
+         xlab = "Salespeople", ylab = "Amount")
> barplot(totP, main = "Transactions per product", names.arg = "",
+         xlab = "Products", ylab = "Amount")
```



آمارهای توضیحی Quant & Val نشان دهنده تغییر پذیری مشخص است . این مسئله بیان می دارد که محصولات ممکن است متفاوت باشند و ممکن است به طور جداگانه تحویل داده شوند . . در حقیقت ، اگر قیمت

های نمونه محصولات خیلی با هم تفاوت داشته باشند گزارش معامله می تواند در مورد همان محصول غیر واقعی باشد . این دو کمیت ممکن است نمونه های ایدهآل نباشند تا نتیجه ای مطلوب حاصل گردد . ارائه کمیت متفاوت از محصولاتی که در هر معامله فروخته می شوند موجب می شود تا به نحو بهتری تحلیل ادامه پیدا کند و قیمت واحد مشخص شود . این قیمت می تواند به ستون جدید چهارچوب اطلاعات ما اضافه شود .

```
> sales$Uprice <- sales$Val/sales$Quant
```

قیمت واحد باید طی معاملات همان محصول ثابت باشند . هنگام تحلیل معاملات طی دوره کوتاه زمانی ، انتظار تغییرات قوی قیمت واحد از محصولات وجود ندارد . اگر ما توزیع قیمت واحد را بررسی نماییم می توانیم مجدداً تغییر پذیری مشخص شده را مشاهده نماییم .

```
> summary(sales$Uprice)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2.4480e-06	8.4600e+00	1.1890e+01	2.0300e+01	1.9110e+01	2.6460e+04
NA's					
1.4136e+04					

با بررسی این حقایق به نظر می رسد که باید مجموعه معاملات هر محصول را به طور جداگانه تحلیل نماییم و در جستجوی معاملات مشکوک در هر مجموعه باشیم . مشکل موجود با این رویکرد محصولاتی هستند که معاملات بسیار پایینی دارند . در عمل ، از تعداد 4548 محصول ، تعداد 982 محصول کمتر از 20 معامله داشته اند . بیان گزارش به عنوان گزارش های غیر واقعی بر اساس نمونه کمتر از 20 گزارش ممکن است بسیار فخرآفرین باشد .

```
> attach(sales)
> upp <- aggregate(Uprice,list(Prod),median,na.rm=T)
> topP <- sapply(c(T,F),function(o)
+               upp[order(upp[,2],decreasing=o)[1:5],1])
> colnames(topP) <- c('Expensive','Cheap')
> topP
```

	Expensive	Cheap
[1,]	"p3689"	"p560"
[2,]	"p2453"	"p559"
[3,]	"p2452"	"p4195"
[4,]	"p2456"	"p601"
[5,]	"p2459"	"p563"

در واقع بررسی محصولاتی که بالاترین و پایین ترین قیمت را دارند می تواند جالب باشد . ما در اینجا از قیمت میانی واحد استفاده نمودیم تا قیمت نمونه محصولی که فروخته شده است را ارائه دهیم . کد زیر اطلاعاتی را که ما جستجو نمودیم بدست آورده است .

ما چهارچوب اطلاعات را ثابت نگه داشتیم تا دسترسی به ستون های اطلاعات هموار شود . سپس قیمت میانی واحد هر محصول را با استفاده از عملکرد `aggregate()` بدست آوردیم . این بخش بیان کننده عملکردی است که برخی مقادیر عددی را برای گروه های فرعی مجموعه اطلاعات فراهم می کند که بر اساس همان فاکتور شکل گرفته است . نتیجه این امر چهارچوب اطلاعات با مقادیر عملکرد انبوه برای هر گروه است . بر اساس این چهارچوب اطلاعات بدست آمده ، ما پنج محصول گرانیقیمت را از طریق کاهش پارامتر دستور `order()` عملکرد ارتقا داده و از عملکرد `supply()` استفاده نمودیم . سپس می توانیم توزیع قیمت مختلف محصولات سطح بالا را با استفاده از ترسیم قیمت های واحد به طور کامل تأیید نماییم

```
> tops <- sales[Prod %in% topP[1, ], c("Prod", "Uprice")]
> tops$Prod <- factor(tops$Prod)
> boxplot(Uprice ~ Prod, data = tops, ylab = "Uprice", log = "y")
```

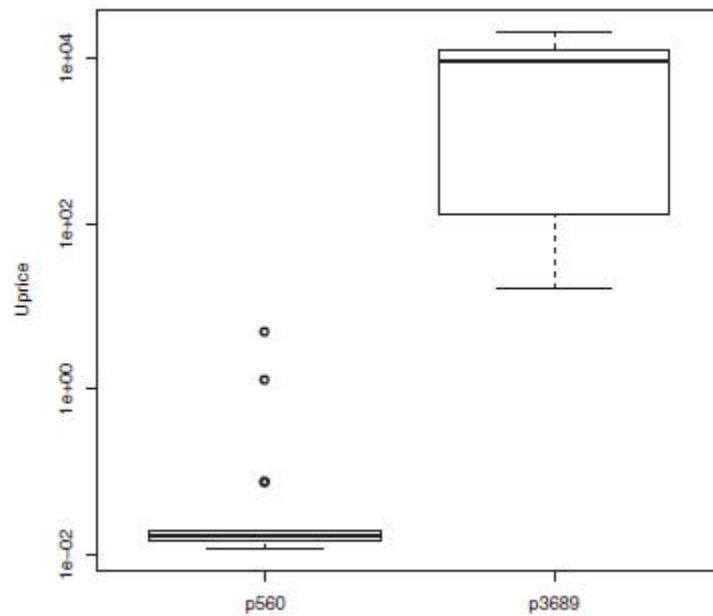
اگر مقدار عملکرد به مجموعه ای تعلق داشته باشد ، اپراتور `%in%` تست می شود . فاکتور عملکرد نیز لازم است چرا که ستون `Prod` از چهارچوب اطلاعات دارای تعدادی سطوح است که به عنوان ستون اصلی چهارچوب اطلاعات فروش در نظر گرفته می شود که منجر به عملکرد `boxplot()` و ترسیم `box plot` برای هر سطح می گردد . مقیاس قیمت ارزانترین و گرانترین محصولات تفاوت هایی دارد . به همین دلیل ، ما از مقیاس ثبت وقایع در نمودار استفاده نمودیم تا از مقادیر ارزان ترین محصولات که غیر قابل تشخیص هستند جلوگیری گردد . این اثر از طریق تنظیم پارامتر `log=y` بدست می آید که نشان می دهد `Y-axis` بر اساس مقیاس ثبت وقایع است . باید توجه داشت که چگونه همان میزان فاصله در محور با مقادیر مختلف قیمت واحد مربوط می گرد . نتیجه این کدبندی در شکل 3-4 نشان داده شده است .

```
> vs <- aggregate(Val,list(ID),sum,na.rm=T)
> scoresSs <- sapply(c(T,F),function(o)
+ vs[order(vs$x,decreasing=o)[1:5],1])
> colnames(scoresSs) <- c('Most','Least')
> scoresSs
```

	Most	Least
[1,]	"v431"	"v3355"
[2,]	"v54"	"v6069"
[3,]	"v19"	"v5876"
[4,]	"v4520"	"v6058"
[5,]	"v955"	"v4515"

ما می توانیم تحلیل مشابه ای ارائه دهیم تا مشخص نماییم که کدام فروشندگان می توانند بیشترین یا کمترین وجه نقد را وارد شرکت کنند . ذکر این نکته جالب است که 100 فروشنده سطح بالا در این لیست در حدود 40% درآمد شرکت را در دست داشتند و از تعداد 6016 فروشنده در سطح پایین تنها 2% درآمد شرکت را در دست داشته اند . این امر فراهم کننده دیدگاه هایی برای تغییرات احتمالی است که در شرکت نیاز می باشد .

شکل 3-4: توزیع قیمت های واحد ارزانترین و گرانترین محصولات



```
> sum(vs[order(vs$x, decreasing = T)[1:100], 2])/sum(Val, na.rm = T) *
+ 100
```

```
[1] 38.33277
```

```
> sum(vs[order(vs$x, decreasing = F)[1:2000], 2])/sum(Val,
+ na.rm = T) * 100
```

```
[1] 1.988716
```

If we carry out a similar analysis in terms of the quantity that is sold for each product, the results are even more unbalanced:

```
> qs <- aggregate(Quant,list(Prod),sum,na.rm=T)
> scoresPs <- sapply(c(T,F),function(o)
+ qs[order(qs$x,decreasing=o)[1:5],1])
> colnames(scoresPs) <- c('Most','Least')
> scoresPs
```

	Most	Least
[1,]	"p2516"	"p2442"
[2,]	"p3599"	"p2443"
[3,]	"p314"	"p1653"
[4,]	"p569"	"p4101"
[5,]	"p319"	"p3678"

```
> sum(as.double(qs[order(qs$x,decreasing=T)[1:100],2]))/
+ sum(as.double(Quant),na.rm=T)*100

[1] 74.63478

> sum(as.double(qs[order(qs$x,decreasing=F)[1:4000],2]))/
+ sum(as.double(Quant),na.rm=T)*100

[1] 8.94468
```

اگر بخواهیم تحلیل مشابهی بر اساس کمیت کالای فروخته شده ارائه دهیم ، نتایج بیشتر غیر متعادل می شوند . شما می توانید به استفاده از عملکرد `as.double()` در بالا توجه داشته باشید . این وضعیت در این نمونه نیاز می باشد چرا که جمع کمیت ها آنقدر بزرگ است که باید به صورت دوتایی ذخیره گردند . چنین عملکردی موجب انتقال می شود .

از تعداد 4548 محصول ، تعداد 4000 محصول کمتر از 10% حجم فروش دارند که رتبه بندی آن در میان 100 محصول رده بالا به مقدار 75% است . باید توجه داشت که این اطلاعات تنها بر حسب تولید محصولات مهم است . به طور ویژه ، منظور این نیست که شرکت باید تولید محصولاتی را که تنها واحدهای بسیار اندکی آنرا می فروشند متوقف سازد بلکه اگر این محصولات بتوانند سود ناخالص بالایی را بهرآه داشته باشند بیشتر مفید خواهند بود . البته ما اطلاعاتی در مورد هزینه های تولید محصولات نداریم و نمی توانیم بر حسب غیر مفید بودن محصولات یا تعداد فروش واحدها نتیجه گیری نماییم . یکی از فرضیات اصلی که در تحلیل ما ایجاد شده است ، یافتن گزارشات معاملات غیر طبیعی است که قیمت واحد هر محصول باید بتواند توزیع نرمالی را برای آن انتخاب نماید . این بدان معناست که ما انتظار داریم معاملات همان محصول دارای همان قیمت واحد با تغییر پذیری کوچک باشد که فروشنده با استفاده از استراتژی هایی توانسته است اهداف تجاری کسب نماید . بر این اساس ، آزمایشات اولیه آماری وجود دارد که می توان از آنها در یافتن مسیرهای انحراف و فرضیات غیر نرمال بهره گرفت . نمونه آن قانون `box plot` است . این قانون به عنوان پایه تعیین موضوعات محسوب می شود و در این کتاب چندین بار به آن اشاره شده است . این قانون بیان می دارد مشاهده باید به عنوان مقدار بالا یا پایین غیر معمول مشخص گردد و میزان خطر بر حسب بالا و پایین بودن تعیین شود :

$Q3 + 1.5 \times IQR$ به طوریکه $Q1$ اولین بخش تقسیم شده (چهار یک) و $Q3$ سومین بخش تقسیم شده است و مقدار تقسیم شده میانی $IQR = (Q3 - Q1)$ است . این قانون ساده به خوبی متغیرها را توزیع می کند و بر اساس آن آمار تعیین می گردد . کد زیر تعیین کننده این موارد است که بر اساس تعریف بالا در هر محصول می باشد.

```
> out <- tapply(Uprice,list(Prod=Prod),
+ function(x) length(boxplot.stats(x)$out))
```

عملکرد `boxplot.stats()` از آمارهای متعدد بدست می آید که در ساخت `box plot` استفاده می شود . در این حالت لیستی از اطلاعات ایجاد می شود . مولفه های این لیست شامل مشاهداتی است که بر اساس قانون `box plot` است .

```
> out[order(out, decreasing = T)[1:10]]
```

```
Prod
p1125 p1437 p2273 p1917 p1918 p4089 p538 p3774 p2742 p3338
376 181 165 156 156 137 129 125 120 117
```

کد بالا نیز تعداد معاملات را برای هر محصول محاسبه می کند . با استفاده از روش بسیار ساده ذکر شده ، تعداد 29446 معامله در نظر گرفته می شود که تقریباً 7% از کل معاملات را شامل می شود .

```
> sum(out)
```

```
[1] 29446
```

```
> sum(out)/nrow(sales) * 100
```

```
[1] 7.34047
```

پرسشی که اینجا مطرح می شود آن است که آیا این قانون ساده برای تعیین موارد خطا کافی است و می تواند کاربردی باشد یا خیر . در بخش 1-4-4-1 ما به ارزیابی این موضوع می پردازیم .

در همین ارتباط نوعی اخطار به برخی نتیجه گیری ها وارد شده است که در این بخش به آنها توجه می شود . با استفاده از اطلاعات به طور مستقل ، گزارشاتی بر مبنای فریبکاری ارائه شده است گرچه شناسایی نگشته اند . این بدان معناست که برخی از این نتیجه گیری ها بر اساس اطلاعات غلط می باشند . مشکل موجود معاملاتی است که اینگونه شکل پیدا می کنند و ما قطعاً نمی دانیم که مقادیر صحیح کدام است . از لحاظ نظری ، تنها معاملاتی می توانند صحیح باشند که ستون `Insp` آن دارای `ok` باشد گرچه تنها 3.6% از اطلاعات اینگونه است . در موقعیت های واقعی نیز این مسئله باید به حساب آورده شود تا شرکت اطلاعات خود را بر پایه خطاها تنظیم ننماید . به دلیل اینکه بازرسی کامل اطلاعات غیر ممکن است ، چنین خطری همواره وجود دارد . در بیشتر موارد ما می توانیم از استفاده مقادیر کوچک معاملاتی که قبلاً در آنها خطا وجود داشته است خودداری نماییم . اقدام دیگر که می توان برای نتایج پیش بینی نشده انجام داد ، توجه به تحلیل نزدیکتر به اطلاعات است که منجر به نتایج شگفت انگیز می شود . این بدان معناست چنین دسته بندی از تحلیل معمولاً "نیازمند تعامل بین کارشناسان است تا کیفیت اطلاعات بررسی شود . اضافه بر آن ، این نوع تحلیل برای اطلاعات با کیفیت پایین اهمیت کلیدی دارد چرا که بسیاری از مشکلات ممکن است بررسی شوند .

مشکلات اطلاعات :

این بخش تلاش دارد تا برخی مشکلات کیفیت اطلاعات که مانعی بر سر راه کاربرد تکنیک ها است بیان گردد .

مقادیر نامشخص :

ما بیان مشکل مقادیر ناشناخته تغییرپذیر کار را شروع می نماییم . همانطور که در بخش 5-2 ذکر گردید ، سه نوع پیشنهاد متناوب وجود دارد :

۱- حذف نمونه ها

۲- تکمیل ناشناخته ها با استفاده از برخی استراتژی ها

۳- استفاده از ابزاری که ارائه دهنده انواع مقادیر است

با در نظر گرفتن ابزارهایی که در این بخش استفاده می شود تنها دو مورد اول برای ما قابل قبول می باشند . همانگونه که قبلاً ذکر گردید ، موضوع اصلی معاملاتی است که دارای مقادیر Quant & Val می باشند . اگر این امر موجب حذف بیشتر معاملات محصول و فروشنده گردد ، حذف همه 888 نمونه مشکل ساز خواهد بود . تعداد کل معاملات هر فروشنده و محصول به صورت زیر بیان می گردد . فروشندگان و محصولات در معاملات مشکل دار هم وجود دارد .

```
> totS <- table(ID)
> totP <- table(Prod)
```

```
> nas <- sales[which(is.na(Quant) & is.na(Val)), c("ID", "Prod")]
```

اکنون فروشنده با سهم بزرگتر معاملات و با ناشناخته ها در Val & Quant بدست می آید . به نظر میرسد که حذف این معاملات معقول خواهد بود و حداقل این وضعیت برای سهم کوچکی از معاملات ارائه می گردد . اضافه بر آن ، تلاش برای تکمیل هر دو ستون خطر بیشتری به همراه دارد . با توجه به محصولات ، این ارقام به صورت زیر هستند .

```
> propS <- 100 * table(nas$ID)/totS
> propS[order(propS, decreasing = T)[1:10]]
```

v1237	v4254	v4038	v5248	v3666	v4433	v4170
13.793103	9.523810	8.333333	8.333333	6.666667	6.250000	5.555556
v4926	v4664	v4642				
5.555556	5.494505	4.761905				

```
> propP <- 100 * table(nas$Prod)/totP
> propP[order(propP, decreasing = T)[1:10]]
```

p2689	p2675	p4061	p2780	p4351	p2686	p2707	p2690
39.28571	35.41667	25.00000	22.72727	18.18182	16.66667	14.28571	14.08451
p2691	p2670						
12.90323	12.76596						

در همین زمینه محصولات متعددی وجود دارد که بیش از 20% از حذف معاملات را در بر دارد . به طور خاص ، محصول p2689 تقریباً 40% از موارد حذف را دارا می باشد . از طرفی دیگر ، اگر ما مقادیر ناشناخته را

تکمیل نماییم ، تنها استراتژی معقول استفاده از اطلاعات در معاملات کامل همان محصول می باشد . این بدان معناست که با استفاده از 60% اطلاعات باقیمانده می توان 40% معاملات محصول را تکمیل نمود . این وضعیت معقولانه تر است . اگر ما به تشابهات بین توزیع قیمت واحد محصولات توجه نماییم متوجه می شویم که این محصولات مشابه محصولات دیگر هستند . بر این اساس، اگر ما نتیجه گیری نماییم که معاملات بسیار کمی پس از حذف شکل گرفته اند پس می توانیم معاملات را با محصولات مشابه مرتبط نماییم و قابلیت اطمینان آماری در شناسایی آزمایش ها افزایش می یابد . به طور خلاصه باید گفت که حذف همه معاملات با مقادیر ناشناخته بر اساس کمیت و مقدار بهترین انتخاب است .

```
> detach(sales)
> sales <- sales[-which(is.na(sales$Quant) & is.na(sales$Val)),]
```

همچنین برای ناتوان ساختن دسترسی مستقیم به ستون های چهارچوب اطلاعاتی ، از عملکرد detach() استفاده نمودیم . دلیل این امر وضعیت عملکرد detach() است . هنگامیکه یک وضعیت همچون attach(sales) را ثبت نماییم ، R هدف جدیدی را برای هر ستون چهارچوب اطلاعات sales فراهم می کند که نسخه هایی از اطلاعات در ستونها قرار دارند . اگر شروع به حذف اطلاعات از چهارچوب اطلاعات sales نماییم این تغییرات نمی توانند اهداف جدید را منعکس نمایند . به طور خلاصه ، هنگامیکه اطلاعات نسبت به تغییرات مستعد می باشند ، عملکرد detach() موجب ایجاد امکاناتی می گردد چرا که مشاهدات ناپایدار اطلاعاتی به پایان می رسند . این مشاهدات شامل مشاهده چهارچوب اطلاعات اصلی و مشاهدات فراهم شده از طریق عملکرد attach() است . بخش بعدی نمایش لحظه ای چهارچوب اطلاعات در زمان تعیین شده است که مهلت آن گذشته است البته اگر چهارچوب اطلاعات پس از فراخوانی detach() اصلاح گردند .

```
> nnasQp <- tapply(sales$Quant,list(sales$Prod),
+                 function(x) sum(is.na(x)))
> propNasQp <- nnasQp/table(sales$Prod)
> propNasQp[order(propNasQp,decreasing=T)[1:10]]
```

p2442	p2443	p1653	p4101	p4243	p903	p3678
1.0000000	1.0000000	0.9090909	0.8571429	0.6842105	0.6666667	0.6666667
p3955	p4464	p1261				
0.6428571	0.6363636	0.6333333				

اکنون اجازه دهید تا باقیمانده گزارشات را با مقادیر نامشخص بر اساس کمیت یا مقدار معاملات مورد تحلیل قرار دهیم . سپس با محاسبه بخشی از معاملات هر محصول که دارای کمیت ناشناخته است شروع می نماییم . در همین ارتباط دو محصول وجود دارد (p2442 , p2443) که با مقادیر و کمیت ناشناخته دارای معاملاتی هستند . بدون اطلاعات بیشتر هر اقدامی غیر ممکن است چرا که محاسبه قیمت واحد به صورت نمونه امکان پذیر نیست . این موارد 54 گزارش هستند و دو مورد از آنها به گزارشات دروغین مربوط می باشند . این امر بدان معناست که بازرسان اطلاعات بیشتری نسبت به مجموعه اطلاعات دارند و به طور حتم با خطاها روبرو شده اند . بر این اساس ما به حذف برخی از آنها می پردازیم .


```
> sales <- sales[!sales$Prod %in% c("p2442", "p2443"), ]
```

متوجه شدید که ما تنها دو محصول را از مجموعه اطلاعات حذف نمودیم و باید سطوح ستون Prod به روز شوند .

```
> nlevels(sales$Prod)
```

```
[1] 4548
```

```
> sales$Prod <- factor(sales$Prod)
```

```
> nlevels(sales$Prod)
```

```
[1] 4546
```

```
> nnasQs <- tapply(sales$Quant, list(sales$ID), function(x) sum(is.na(x)))
```

```
> propNasQs <- nnasQs/table(sales$ID)
```

```
> propNasQs[order(propNasQs, decreasing = T)[1:10]]
```

v2925	v5537	v5836	v6058	v6065	v4368	v2923
1.0000000	1.0000000	1.0000000	1.0000000	1.0000000	0.8888889	0.8750000
v2970	v4910	v4542				
0.8571429	0.8333333	0.8095238				

همانطور که مشاهده نمودید ، چندین فروشنده وجود دارند که در اطلاعات و کمیت در گزارشات تکمیل نشده اند . به هرحال ، مشکل آنقدر هم پیچیده نمی باشد . در حقیقت ، بر اساس معاملات محصولات گزارش شده توسط دیگر فروشندگان ما می توانیم از این اطلاعات استفاده نموده و موارد ناشناخته را با فرضیات قیمت واحد و موارد مشابه تکمیل نماییم . پس ما این معاملات را حذف نمی نماییم . سپس تحلیل مشابهی را برای معاملات با مقدار مشخص و در ستون Val انجام می دهیم . ابتدا بخشی از معاملات هر محصول با مقدار نامشخص به صورت زیر است .

```
> nnasVp <- tapply(sales$Val, list(sales$Prod),
```

```
+ function(x) sum(is.na(x)))
```

```
> propNasVp <- nnasVp/table(sales$Prod)
```

```
> propNasVp[order(propNasVp, decreasing=T)[1:10]]
```

p1110	p1022	p4491	p1462	p80	p4307
0.25000000	0.17647059	0.10000000	0.07500000	0.06250000	0.05882353
p4471	p2821	p1017	p4287		
0.05882353	0.05389222	0.05263158	0.05263158		

ارقام بسیار منطقی هستند بنابراین ، می توان معاملاتی را با استفاده از معاملات دیگر حذف کرد . با توجه به فرد فروشنده ، ارقام به صورت زیر است .

```
> nnasVs <- tapply(sales$Val, list(sales$ID), function(x) sum(is.na(x)))
> propNasVs <- nnasVs/table(sales$ID)
> propNasVs[order(propNasVs, decreasing = T)[1:10]]
```

```
      v5647      v74      v5946      v5290      v4472      v4022
0.37500000 0.22222222 0.20000000 0.15384615 0.12500000 0.09756098
      v975      v2814      v2892      v3739
0.09574468 0.09090909 0.09090909 0.08333333
```

در این مرحله ما همه گزارشاتی که اطلاعات ناکافی داشتند حذف نمودیم . برای باقیمانده مقادیر ناشناخته از روشی استفاده می نماییم که بر اساس فرضیه هستند و معاملات همان محصول باید دارای قیمت واحد مشابه باشند . ما از روی قیمت های معاملات که دروغین بودن آنها در محاسبات مشخص گردید پرش می نماییم . برای باقیمانده معاملات از قیمت میانی واحد استفاده می نماییم که به عنوان قیمت نمونه محصولات محسوب می شوند .

```
> tPrice <- tapply(sales[sales$Insp != "fraud", "Uprice"],
+   list(sales[sales$Insp != "fraud", "Prod"]), median, na.rm = T)
```

با دارا بودن قیمت واحد نمونه برای هر محصول ما می توانیم از آن برای محاسبه دو مقدار مفقود شده استفاده نماییم (Quant & Val) . این مسئله محتمل است چرا که با دو مقدار مفقود شده هیچ معامله ای وجود ندارد . کد زیر در تمام مقادیر ناشناخته باقیمانده تکمیل می شود .

```
> noQuant <- which(is.na(sales$Quant))
> sales[noQuant,'Quant'] <- ceiling(sales[noQuant,'Val'] /
+   tPrice[sales[noQuant,'Prod']])
> noVal <- which(is.na(sales$Val))
> sales[noVal,'Val'] <- sales[noVal,'Quant'] *
+   tPrice[sales[noVal,'Prod']]
```

در نمونه مفقود شده ، 12900 مقدار کمیت ناشناخته به اضافه 294 مقدار کل معامله تکمیل می شود . همچنین از عملکرد ceiling() استفاده می شود تا از مقادیر غیر مرتبط Quant جلوگیری شود . این عملکرد موجب می شود تا کوچکترین مقدار بازگردد .

با توجه به مطالب بیان شده ما اکنون دارای همه مقادیر Quant & Val هستیم . پس می توانیم مجدداً "ستون Uprice را برای تکمیل قیمت های ناشناخته واحد استفاده نماییم .

```
> sales$Uprice <- sales$Val/sales$Quant
```

بعد از همه مراحل پیش پردازش ، اکنون ما دارای مجموعه اطلاعات آزاد از مقادیر ناشناخته هستیم برای تحلیل بیشتر باید این وضعیت فعلی حفظ گردد . شما نیز می توانید مجدداً "تحلیل خود را از این نقطه آغاز نمایید بدون اینکه تمام مراحل تکرار شوند .

```
> save(sales, file = "salesClean.Rdata")
```

عملکرد `save()` را می توان برای ذخیره کردن مجموعه اهداف در فابل مشخص و در پارامتر `file` مورد استفاده قرار داد . سپس اهداف در این فایل ها ذخیره می شوند تا بتوانند هنگام بارگذاری به `R` و عملکرد `load()` بازگردند .

معاملات بسیار اندک برخی محصولات :

در همین ارتباط محصولاتی هستند که معاملات بسیار کم برای آنها وجود دارد . این نیز مشکلی است چرا که ما نیاز داریم تا از اطلاعات این معاملات بهره برده و در صورت غیر طبیعی بودن متوجه آنها شویم . اگر تعداد معاملات کم باشد ، تصمیم گیری در مورد آنها و بررسی وضعیت آماری آنها بسیار مشکل است . پرسشی که در اینجا مطرح می شود این است که چگونه می توانیم معاملات برخی محصولات را آنالیز نماییم تا از بروز مشکل جلوگیری نماییم .

علیرغم فقدان کامل اطلاعات در ارتباطات موردی بین محصولات ، ما می توانیم تلاش نماییم که تشابهاتی بین برخی از این ارتباطات و توزیع قیمت واحد جستجو نماییم . اگر محصولاتی با قیمت مشابه یافتیم می توانیم وجود معاملات مشابه را در آنها یافته و برای جستجوی مقادیر غیر معمول آنها را آنالیز نماییم . یک روش مقایسه دو توزیع ، بازبینی آنها است . با ارائه تعداد محصول می توان ویژگیهای آماری را در توزیع خلاصه نمود . دو ویژگی مهم توزیع متغیرهای متناوب ، گرایش مرکزی و گسترش آن است . همانطور که قبلاً بیان گردید بسیار معقول است تا تصور نماییم توزیع قیمت واحد هر محصول طبیعی است . در واقع اگرچه تغییر پذیری در قیمت رخ داده است اما آنها باید به بهترین شکل تنظیم گردند . به هر حال ، ما باید تصور نماییم که مقادیر بیان شده چف میزان مشکل دارند و آیا موارد خطا و تقلب وجود دارد یا خیر . استفاده از ابزار میانی به عنوان بررسی آماری و `IQR` می تواند مناسب باشد . این آمارها مهم هستند و می توان برای نمونه هایی که در مسیر منحرف هستند از آنها استفاده نمود . همچنین ما می توانیم از آمارها برای تمام معاملات و برای هر محصول استفاده نماییم .

```
> attach(sales)
> notF <- which(Insp != 'fraud')
> ms <- tapply(Uprice[notF],list(Prod=Prod[notF]),function(x) {
+   bp <- boxplot.stats(x)$stats
+   c(median=bp[3],iqr=bp[4]-bp[2])
+ })
> ms <- matrix(unlist(ms),
+               length(ms),2,
```



```

+ byrow=T,dimnames=list(names(ms),c('median','iqr'))))
> head(ms)

      median      iqr
p1 11.346154 8.575599
p2 10.877863 5.609731
p3 10.000000 4.809092
p4  9.911243 5.998530
p5 10.957447 7.136601
p6 13.223684 6.685185

```

این کد از عملکرد `boxplot.stats()` استفاده می نماید تا مقادیر مختلف را کسب نماید . ما این مقادیر را برای همه مجموعه معادلات در هر محصول محاسبه نمودیم و معاملات دروغین و متقلبانه را از تحلیل خود رفع نمودیم . با این مقادیر، ما توانستیم برای هر محصول یک ماتریکس با مقدار میانگین و IQR بدست آوریم . شکل 4-4 الف نشان دهنده وضعیت هر محصول بر اساس میانگین و IQR است . خواندن نمودار موجود مشکل است چرا که تعدادی از آنها برای این آمارها دارای مقادیر بسیار بزرگ هستند . به طور ویژه ، محصول p3689 با محصولات دیگر شرکت متفاوت است و ما می توانیم با استفاده از مقیاس های ثابت وقایع بر این مشکل فائق آییم . در نمودار دوم از علامت + مشکی استفاده شده است که نشان دهنده محصولاتی است که کمتر از 20 معامله دارند . نمودار به صورت زیر بدست می آید به طوریکه ب پارامتر `log=xy` در مجموعه های هر دو محور از نمودار به صورت زیر است.

```

> par(mfrow = c(1, 2))
> plot(ms[, 1], ms[, 2], xlab = "Median", ylab = "IQR", main = "")
> plot(ms[, 1], ms[, 2], xlab = "Median", ylab = "IQR", main = "",
+      col = "grey", log = "xy")
> smalls <- which(table(Prod) < 20)
> points(log(ms[smalls, 1]), log(ms[smalls, 2]), pch = "+")

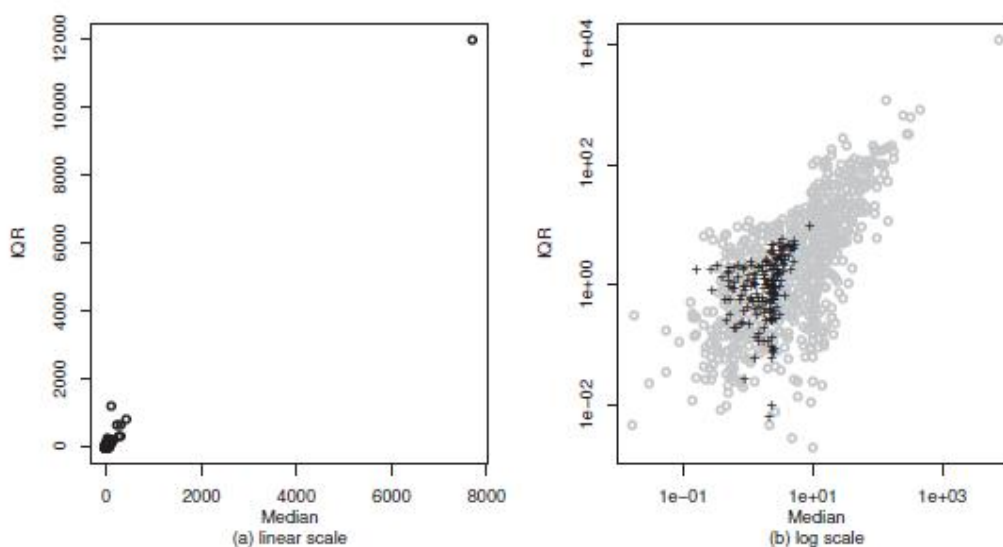
```

در شکل 4-4 ب متوجه می شویم که محصولات زیادی وجود دارند که به طور تقریبی دارای همان میزان میانگین و IQR هستند و حتی هنگامی که به مقیاس های بزرگ توجه می کنیم آنها را به حساب می آوریم . این امر تشابه توزیع قیمت واحد را مشخص می نماید . اضافه بر آن ، ما می توانیم در میان محصولات معاملات بسیار اندکی را ببینیم که با دیگر محصولات تشابه داند . به هر حال ، محصولات مختلف نه تنها معاملات اندکی دارند بلکه توزیع قیمت واحد در آنها فرق دارد . در همین ارتباط محصولاتی هستند که در معاملات آنها تقلب مشاهده شده است .

علیرغم مزایای بازرسی بصری ویژگیهای توزیع قیمت های واحد ، آزمایشات رسمی نیاز است تا هنگام مقایسه توزیع بین دو محصول دقت بیشتری حاصل گردد . در همین ارتباط ما از آزمایش غیر پارامتریک استفاده می نماییم تا توزیع قیمت های واحد را مقایسه نماییم . این آزمایشات نسبت به وجود خطاها قوی هستند . آزمایش

Kolmogorov – Smirnov برای هر دو نمونه شرکت استفاده می شود تا صحت فرضیات در همان توزیع

بررسی گردد .



این آزمایش با محاسبه آمار انجام می گیرد و به ارزیابی ماکزیمم تفاوت بین دو عملکرد توزیع تجربی می پردازد . اگر دو توزیع مشابه باشند این فاصله کوچک است . برای هر کدام از محصولات که کمتر از 20 معامله دارند ما هر محصول را در توزیع قیمت واحد مشابه جستجو می کنیم و از آزمایش Kolmogorov – Smirnov برای بررسی استفاده می نماییم البته اگر تشابهات از لحاظ آماری قابل توجه باشند . انجام این فعالیت برای همه گروه های محصول قابل محاسبه است . بر این اساس ما تصمیم گرفتیم تا امتیاز اطلاعات کسب شده را بر اساس ویژگیهای توزیع شده به حساب آوریم . برای هر کدام از محصولاتی که تنها معاملات کمی دارند به جستجو پرداختیم تا محصولی با شباهت بیشتر به IQR بیابیم . با توجه به محصول مشابه ما آزمایش Kolmogorov – Smirnov را بین توزیع قیمت واحد دنبال نمودیم و نتایج آزمایش را بررسی نمودیم . کد حاصله ، ماتریکس مشابهی با اطلاعات این نوع آزمایش بدست می آورد که برای هر محصول با کمتر از 20 معامله است . همچنین از ms استفاده نمودیم .

```
> dms <- scale(ms)
> smalls <- which(table(Prod) < 20)
> prods <- tapply(sales$Uprice, sales$Prod, list)
> similar <- matrix(NA, length(smalls), 7, dimnames = list(names(smalls),
+   c("Simil", "ks.stat", "ks.p", "medP", "iqrP", "medS",
+     "iqrS")))
> for (i in seq(along = smalls)) {
```

```

+ d <- scale(dms, dms[smalls[i], ], FALSE)
+ d <- sqrt(drop(d^2 %*% rep(1, ncol(d))))
+ stat <- ks.test(prods[[smalls[i]]], prods[[order(d)[2]]])
+ similar[i, ] <- c(order(d)[2], stat$statistic, stat$p.value,
+ ms[smalls[i], ], ms[order(d)[2], ])
+ }

```

کد از طریق نرمال سازی اطلاعات بر اساس ms شروع می گردد تا هنگام محاسبه فواصل از اثرات منفی جلوگیری شود. پس از تعدادی فرمت بندی ما دارای حلقه اصلی هستیم که همه محصولات با تعداد محدودی معامله منتقل می گردد. دو وضعیت اولیه در این بخش به محاسبه فواصل بین ویژگیهای توزیع می پردازد. نتیجه هدف d دارای مقادیر همه این فواصل است. کوچکترین فاصله محصولی است که مشابه با محصولات در نظر گرفته شده است. باید توجه داشت که تشابه بین محصولات با استفاده از اطلاعات IQR قیمت های واحد محاسبه می شود. مرحله بعدی، انتقال آزمایش Kolmogorov – Smirnov است که دو توزیع قیمت های واحد مقایسه گردد. این عملکرد بوسیله ks.test() انجام می گیرد. این عملکرد اطلاعات بارزی دارد و مقادیر آن از لحاظ آماری تست می شوند. مقدار آمار حداکثر تفاوت بین دو عملکرد توزیعی است. مقادیر سطح اطمینان نزدیک 1 نشان دهنده وضعیت آماری قوی است که هر دو توزیع با هم مساوی می باشند. در جدول ارائه شده، نام های ردیف نشان دهنده محصول است که ما بیشترین تشابه را بدست آوردیم. ستون اول دارای اطلاعاتی می باشد. ID محصول نیز به صورت زیر محاسبه می شود.

```
> head(similar)
```

	Simil	ks.stat	ks.p	medP	iqrP	medS	iqrS
p8	2827	0.4339623	0.06470603	3.850211	0.7282168	3.868306	0.7938557
p18	213	0.2568922	0.25815859	5.187266	8.0359968	5.274884	7.8894149
p38	1044	0.3650794	0.11308315	5.490758	6.4162095	5.651818	6.3248073
p39	1540	0.2258065	0.70914769	7.986486	1.6425959	8.080694	1.7668724
p40	3971	0.3333333	0.13892028	9.674797	1.6104511	9.668854	1.6520147
p47	1387	0.3125000	0.48540576	2.504092	2.5625835	2.413498	2.6402087

```
> levels(Prod)[similar[1, 1]]
```

```
[1] "p2829"
```

پس از بررسی ستون ها با سطح اطمینان Kolmogorov – Smirnov، ما می توانیم چگونگی وضعیت محصول را بر اساس توزیع قیمت واحد و تشابه آن با سطح اطمینان 90% بررسی نماییم.

```
> nrow(similar[similar[, "ks.p"] >= 0.9, ])
```

```
[1] 117
```

Or more efficiently,

```
> sum(similar[, "ks.p"] >= 0.9)
```

```
[1] 117
```

با توجه به 985 محصول و تنها 20 معامله آن، ما نیز تنها محصولات مشابهی برای 117 مورد یافتیم. این اطلاعات هنگامی مناسب هستند که معاملات غیر طبیعی اعلام شوند. برای 117 محصول ما می توانیم معاملات بیشتری را بر اساس فرایند برنامه ریزی تعیین کنیم تا وضعیت آماری را افزایش دهیم.

```
> save(similar, file = "similarProducts.Rdata")
```

تعریف فعالیت های استخراج اطلاعات :

هدف اصلی این اقدام استفاده از اطلاعات استخراج شده و ایجاد راهنمایی در مورد تصمیم گیری روی گزارشات معاملاتی که احتمال فریبکاری در آنها وجود دارد و باید مورد بازبینی و بررسی قرار بگیرند. با توجه به محدودیت منابع دردسترس جهت بررسی معاملات، چنین راهنمایی باید شکلی از رتبه بندی میزان احتمال فریبکاری را مشخص نماید.

رویکردهای مختلف در زمینه مشکل : مجموعه اطلاعات دردسترس دارای ستون **Insp** است که دارای اطلاعاتی در زمینه فعالیت های بازرسی قبلی دارد. مشکل اصلی این است که اکثریت گزارشات دردسترس مورد بررسی قرار نگرفته اند. با توجه به این مطلب هیچ معامله ای دروغین اعلام نشده است و اصطلاح **unkn** در متغیر **Insp** به معنای متغیر ناشناخته است. این مقدار نشان دهنده فقدان اطلاعات مناسب در مورد معاملات است. این بدان معناست که ما دارای دو نوع مشاهده در مجموعه اطلاعات هستیم. اول اینکه مجموعه کوچک دارای توضیحاتی در مورد ویژگیهای مشاهدات و نتایج بازرسی شده است. دوم، مجموعه بزرگ که مشاهدات بازبینی نشده را نشان می دهد و در ستون **unkn** بیشترین مقدار را دارد. در همین ارتباط انواع مختلف مدلسازی رویکرد وجود دارد که ما می توانیم از آنها استفاده نماییم.

تکنیک های غیر نظارتی :

در گزارشاتی که بازبینی صورت نگرفته است ستون Insp با هیچگونه اطلاعاتی مرتبط نمی باشد . جهت چنین مشاهداتی ما تنها افرادی را در دسترس داریم که معاملات را شرح می دهند . این بدان معناست که گزارشات فروش تنها بوسیله مجموعه ای از متغیرهای مستقل شرح داده شده اند . این نوع اطلاعات از طریق تکنیک های یادگیری استفاده می شوند . چنین روش هایی نام خاصی ندارند چرا که هدف آنها آموختن مفاهیم توسط معلم به عنوان روش نظارتی نیست . اطلاعات مورد استفاده برای اینگونه روش ها نیازمند پیش طبقه بندی است تا کارشناس مجرب بتواند مبانی هدف را دنبال نماید . ما در این ارتباط با فعالیت استخراج اطلاعات توضیحی مواجه شده ایم که هدف روش های نظارتی است.

دسته بندی نمونه ای از تکنیک استخراج اطلاعات توضیحی می باشد . روش های دسته بندی تلاش دارند تا گروه های طبیعی مجموعه مشاهدات را با استفاده از شکل دادن دسته ها و نمونه های مشابه پیدا کنند . موضوع تشابه معمولاً "نیازمند تعریف متریک بر عملکرد فاصله است که چگونگی مشاهدات را ارزیابی می نماید . نمونه هایی که نزدیک یکدیگر هستند و معمولاً" بخشی از همان گروه اطلاعاتی در نظر گرفته می شوند . شناسایی خطاها به عنوان فعالیت استخراج اطلاعات توصیفی در نظر گرفته می شود . برخی از روش های شناسایی توزیع مشخصی از اطلاعات را به همراه دارند . استراتژی شناسایی خطا روشی متریک محسوب می شود که به متغیرهایی توجه دارد .

بر اساس مباحث بالا ما می توانیم ارتباطات قوی بین دسته بندی و شناسایی خطاها را بررسی نماییم . این مسئله نوعی روش شناسی محسوب می شود که بر اساس وضعیت بین مشاهدات است . خطاها نمونه های مختلفی هستند و باید به درستی دسته بندی شوند . هدف از دسته بندی مجموعه اطلاعات تنها نباید به گروه های بزرگ ختم شود . استفاده از تکنیک های غیر نظارتی در مشکل ما شامل برخی محدودیت ها است . در حقیقت هدف ما کسب رتبه بندی برای مجموعه مشاهدات است . این رتبه بندی پایه ای برای تصمیمات بازبینی محسوب می شود . این بدان معناست که ابزارهایی که ما انتخاب نمودیم قادرند خطاها را مشخص کنند و بعد به رتبه بندی آنها بپردازند . در بخش 1-4-4 به تکنیک های غیر نظارتی که برای بیان فعالیت استخراج اطلاعات انتخاب شده اند اشاره می نماییم .

تکنیک های نظارتی :

مجموعه معاملاتی که پس از بازرسی عناوین " درست " یا " جعلی " را کسب کرده اند می توانند با دیگر رویکردهای مدلسازی استفاده شوند . روش های یادگیری نظارتی از این نوع اطلاعات بهره می برند . هدف از این رویکردها ، کسب مدل هایی مرتبط با متغیر هدف است که مجموعه ای از متغیرهای مستقل در آن وجود دارد . این مدل به عنوان تقریب بین عملکرد ناشناخته $Y = f(X_1, X_2, \dots, X_p)$ است که ارتباط بین متغیر هدف Y و پیش بینی های X_1, X_2, \dots, X_p را شرح دهد . تکنیک مدلسازی برای بدست آوردن پارامترهایی است که معیار مشخص انتخاب شده را بهینه می سازد و خطای مدل را به حداقل می رساند . چنین فعالیت تحقیقی با کمک مشاهدات انجام می گیرد و بر اساس مجموعه اطلاعاتی است که شامل نمونه هایی از مفاهیم یادگرفته شده است . این نمونه ها در واقع نمونه هایی خاص از X_1, X_2, \dots, X_p است . اگر متغیر هدف Y متناوب باشد

می توانیم ترکیبی از مشکل داشته باشیم و اگر متغیر Y به صورت عددی باشد ما طبقه بندی از مشکل خواهیم داشت .

در نمونه پایگاه اطلاعاتی ، متغیر هدف نتیجه فعالیت بازبینی است و می تواند دو مقدار ممکن را بدست آورد : OK , Fraud . این بدان معناست که هدف ما یادگیری مفاهیم گزارشات جعلی و واقعی است .

آنچه که ما با آن روبرو هستیم مشکل طبقه بندی است . باید توجه داشت معاملاتی که بازبینی نشده اند قابل استفاده در فعالیت ها نیستند چرا که هنوز مشخص نیست که این فعالیت ها و معاملات جعلی هستند و یا واقعی و صحیح هستند . پس اگر ما بخواهیم از تکنیک طبقه بندی استفاده نماییم باید با استفاده از گزارش نوعی مدلسازی نماییم . ما تنها می توانیم تعداد 15732 نمونه از 401146 گزارش نمونه به عنوان نمونه بررسی شده بدست آوریم .

مشکل طبقه بندی که با آن روبرو هستیم می تواند روی ارزیابی عملکرد مدلها اثر بگذارد . چنین وضعیتی در حقیقت در میان دو مقدار طبقه بندی ممکن شکل می گیرد . از تعداد 15732 گزارش بازبینی شده تعداد 14462 معامله طبیعی و تنها 1270 نمونه دروغین بوده اند . اضافه بر آن ، هدف اصلی این کار شناسایی موارد جعلی است . این بدان معناست که ما باید معیارهایی را برای ارزیابی انتخاب نماییم تا بتوانیم عملکرد مدل ها را تعیین نماییم و با انتخاب تکنیک های مدلسازی بتوانیم بر مشکلات مجموعه اطلاعات و عدم تعادل ها غلبه نماییم .

استفاده از ابزارهای طبقه بندی شامل برخی وفق پذیری ها است . در این راستا ما علاقمند به کسب رتبه بندی معاملات بر اساس احتمال جعلی بودن هستیم . همچنین با استفاده از مجموعه آزمایش گزارشات و استفاده از مدل ها می توانیم نمونه ها را بازرسی نماییم . تعدادی از الگوریتم های طبقه بندی تنها می توانند خروجی ها را بررسی کنند . چنین وضعیتی برای رفع مشکل ما کافی نیست چرا که ما نیاز به ابزار یا روش های داریم تا با ایجاد رتبه بندی نمونه های طبقه بندی شده بتوانیم نمونه گزارشات معاملات جعلی را مشخص نماییم . آنچه که ما نیاز داریم طبقه بندی احتمالی است چرا که مدلسازی نه تنها می تواند نمونه ها را پیش بینی نماید بلکه میزان احتمال را نیز مشخص می نماید . این میزان احتمالات به ما این امکان را می دهند تا نمونه های آزمایش شده را بر اساس وضعیتشان رتبه بندی نماییم .

تکنیک های نیمه نظارتی :

روش های نیمه نظارتی از طریق مشاهدات ایجاد می شوند که کاربردهای متعددی در یافتن اطلاعات دارند ؛ نمونه هایی که دارای مقدار تعیین شده ای متغیر هدف می باشند . معمولاً این اطلاعات نیازمند فعالیت کارشناسان است و موجب افزایش هزینه های جمع آوری اطلاعات می گردد . از طرف دیگر ، کسب اطلاعات غیر مشخص آسان است به ویژه اینکه استفاده گسترده ای از سنسورها و دیگر انواع ابزارهای جمع آوری اطلاعات اتوماتیک می گردد .

روش های نیمه نظارتی نیز در این بخش قابل استفاده هستند چرا که می توانند این نوع مجموعه اطلاعات را در نمونه های مشخص و غیر مشخص ارائه دهند . معمولاً دو نوع روش نیمه نظارتی وجود دارد . از یک طرف روش های طبقه بندی نیمه نظارتی که تلاش بر بهبود عملکرد الگوریتم های طبقه بندی استاندارد نظارتی دارد و در کسب اطلاعات از نمونه های غیر مشخص کمک می نماید . رویکرد متناوب در واقع روش های طبقه بندی نیمه نظارتی بر اساس اطلاعات مشخص جهت معیارهایی لازم برای شکل گیری گروه ها است .

در گروه بندی یا دسته بندی نیمه نظارتی ، ایده اصلی بر سر استفاده از اطلاعات در دسترس برای فرایند دسته بندی است که شامل نمونه هایی با همان اطلاعات و همان گروه می باشد و یا نمونه هایی با اطلاعات مختلف در گروه های مختلف است . معایر مورد استفاده جهت شکل گیری دسته ها ، تغییر روش ها و یافتن گروه های مناسب برای نمونه ها است . بر اساس رویکردهای نیمه نظارتی مشابه ، استفاده از الگوریتم ها می تواند موجب بهینه سازی محدودیت ها گردد .

با توجه به طبقه بندی نیمه نظارتی ، روش شناسی های مختلفی وجود دارد . یکی از روش های شناخته شده ، خود آموزی است . این رویکرد نوعی رویکرد تعاملی است . مرحله بعد استفاده از مدل برای طبقه بندی اطلاعات نامشخص است . نمونه هایی که مدل میزان اطمینان بیشتر دارد به اطلاعات نامشخص تقسیم می شوند . با استفاده از مجموعه جدید مدل جدیدی بدست می آید و مورد پردازش قرار می گیرد تا برخی معیارها حاصل گردند . از نمونه های مدل های طبقه بندی نیمه نظارتی می توان به TSVMs اشاره کرد . هدف از TSVMs کسب نمونه هایی برای مجموعه اطلاعات نامشخص است .

مجدداً باید محدودیت های خاص کاربردها را در نظر بگیریم که بر حسب رتبه بندی می باشند . این امر موجب می شود تا از استراتژی های لازم برای روش های نظارتی و غیر نظارتی استفاده شود که این مسئله به طبقه بندی های اطلاعات مربوط می شود .

ارزیابی معیارها :

در این بخش ما به ارزیابی مدل ها می پردازیم . هنگامیکه مجموعه آزمایش گزارشات معاملات ارائه می شود ، هر کدام از مدل ها اقدام به ایجاد رتبه بندی گزارشات می نمایند . این بخش به شرح این رتبه بندی اشاره دارد . همچنین در این بخش از روش شناسی تجربی استفاده می شود که برای تخمین متریک های ارزیابی انتخابی لازم است . مجموعه اطلاعات ما به طور ویژه شامل اطلاعات مشخص و نامشخص است . بر این اساس دو موقعیت در گزارشات بازرسی شده و بازرسی نشده وجود دارد و همین امر موجب افزایش مقایسه مدل ها می شود . رتبه بندی ها بوسیله مدل هایی ایجاد می شود که بر اساس مشاهدات مشخص و نامشخص هستند . در واقع ارزیابی گزارشات به نوع اطلاعات مربوط می شود و در پاره ای از موارد ارزیابی بسیار سخت است چرا که نمی توان جعلی بودن و جعلی نبودن را مشخص کرد .

دقت و فراخوانی :

بر اساس این نوع کاربرد ، مدل موفقیت آمیزی بر اساس رتبه بندی بدست می آید که شامل همه نمونه گزارشات جعلی شناخته شده در موقعیت های بالای رتبه بندی است . گزارشات دروغین در اطلاعات ما در

اقلیت قرار دارند . با وجود رقم k در گزارشاتی که منابع ما اجازه بازنگری را می دهند ، علاقمند هستیم که بالاترین موقعیت k را در رتبه بندی حاصله داشته باشیم . اضافه بر آن ، ما می خواهیم موقعیت های k را در تمام نمونه های شناخته شده جعلی که در مجموعه آزمایشات ما وجود دارند بررسی نماییم .

همانطور که در بخش 3-3-4 مشاهده نمودید ، هنگامی که هدف ما پیش بینی مجموعه کوچک از رخدادهای کمیاب است ، دقت و فراخوانی نوعی اندازه گیری در ارزیابی مناسب محسوب می شود . با توجه به محدودیت بازبینی k ، ما می توانیم دقت و بازخوانی بالاترین موقعیت k را در رتبه بندی محاسبه نماییم . این مقدار یا حد k تعیین کننده این موضوع است که کدام گزارشات بر اساس مدلسازی مورد بازبینی قرار گرفته اند . با توجه به دیدگاه طبقه بندی شده موقعیت k در موارد جعلی قابل شناسایی است به طوریکه باقیمانده گزارشات طبیعی هستند . مقدار دقت به ما می گوید که چه میزان گزارش موقعیت k جعلی است . مقدار فراخوانی به ارزیابی تعداد گزارشات جعلی در مجموعه آزمایشات می پردازد . باید به این نکته توجه نمود که مقادیر حاصله نوعی بدبینی را نشان می دهد . در حقیقت اگر موقعیت k شامل گزارشات غیر مشخص باشد ما نمی توانیم دقت و فراخوانی را محاسبه نماییم . به هر حال ، اگر گزارشات بازنگری شوند می توانیم مقادیر واقعی را پیدا نماییم . معمولاً بین دقت و فراخوانی نوعی رابطه جایگزینی وجود دارد . مثلاً " کاملاً " آسان است تا 100 فراخوانی را کسب کنیم البته اگر همه نمونه های آزمایش شده بتوانند رخدادها را پیش بینی کنند .

به هر حال ، چنین استراتژی به ضرورت منجر به دقت پایین می شود . کاربرد فعلی دارای جزئیات متعددی است . با توجه به این حقیقت که محدودیت هایی در منابع سرمایه گذاری شده در فعالیت های بازنگری وجود دارد ، آنچه که ما واقعاً خواستار آن هستیم افزایش استفاده از منابع است . این بدان معناست که اگر ما X ساعت جهت بازنگری گزارشات صرف نماییم و اگر X ساعت گزارش طبیعی بازنگری شود آنگاه در رتبه بندی ما دقت پایین است . فراخوانی دقیقاً موضوع اصلی در این بخش است و آنچه که ما قادریم تا کسب نماییم در واقع 100% منابع در دسترس است .

نمودارهای بالابرنده و دقت / منحنی های فراخوانی :

در بخش پیشین در مورد محاسبات مقادیر دقت و فراخوانی جهت بازنگری های انجام شده مطالبی ارائه شد . جالب است تا عملکرد مدل ها در سطوح مختلف بررسی شود و مدل های مختلف در سطوح مختلف آشکار شوند و این موضوعات هنگام مقایسه اطلاعات مناسبی محسوب می گردد .

منحنی های دقت / فراخوانی (PR) از نمونه های بصری عملکرد مدل بر حسب آمارهای دقت و فراخوانی است . منحنی ها از طریق مقادیر آماری در نقاط مختلف بدست می آیند . این نقاط از طریق حد های برش در رتبه بندی نمونه ها حاصل می شود . در نمونه ما ، این مورد به حدهای مختلف مربوط می شود که برای رتبه بندی خطا در مدل ها بکار می رود . بر این اساس ، مقادیر مختلفی از دقت و فراخوانی استفاده نمودیم . منحنی PR به ما امکان این نوع تحلیل را میدهد .

بسته ROCR شامل عملکرد های مختلف است که بسیار مناسب ارزیابی طبقه بندی های دوتایی است . این امر نوعی بسته بندی فوق العاده می باشد که شما باید قبل از تلاش کد زیر نصب گردد . بسته مورد نظر بسیاری از

ارزیابی ها را تحقق می بخشد و شامل روش هایی جهت کسب منحنی های وسیع است . استفاده از این نوع بسته بندی ساده است . ما با بدست آوردن هدف از نوع طبقه `prediction` و استفاده از پیش بینی ها در مدل و مقادیر صحیح مجموعه آزمایش کار خود را شروع نمودیم . فعالیت انجام شده، عملکرد `(prediction)` است . نتیجه این امر نیازمند متریک ارزیابی مختلف بر اساس عملکرد `(performance)` است . سرانجام ، نتیجه عملکرد نهایی با عملکرد `plot()` نیازمند منحنی های مختلف عملکرد است . کد زیر نمایشی از این فرایند با استفاده از اطلاعات نمونه می باشد .

```
> library(ROCR)
> data(ROCR.simple)
> pred <- prediction(ROCR.simple$predictions, ROCR.simple$labels)
> perf <- performance(pred, "prec", "rec")
> plot(perf)
```

این کد ، منحنی PR را رسم می نماید که در شکل 5.4 در نمودار سمت چپ نشان داده شده است . منحنی های مذکور توسط بسته POCR ایجاد گشته که دارای شکل خاصی است . گرچه این وضعیت کاملاً شفاف نیست و روش هایی برای غلبه بر این مسئله وجود دارد . ما می توانیم دقت را برای سطحی از فراخوانی مشخص محاسبه نماییم و بالاترین میزان دقت را برای هر سطح فراخوانی بیابیم که بزرگتر یا مساوی r است.

$$Prec_{int}(r) = \max_{r' \geq r} Prec(r') \quad (4.1)$$

اگر نگاه دقیقی به نتیجه داشته باشیم می توانیم عملیات `(performance)` را بررسی نماییم که به صورت `y.values` با مقادیر محور `y` در نمودار مشخص شده است و مقدار دقت نیز رسم شده است . همچنین می توانیم منحنی PR را بدون اثر خاصی بدست آوریم که با جایگزینی ساده مقادیر و بر اساس معادله 4-1 حاصل می شود . عملیات زیر می تواند این ایده را برای نمونه های کلی تحقق بخشد .

```
> PRcurve <- function(preds, trues, ...) {
+   require(ROCR, quietly = T)
+   pd <- prediction(preds, trues)
+   pf <- performance(pd, "prec", "rec")
+   pf@y.values <- lapply(pf@y.values, function(x) rev(cummax(rev(x))))
```

کد مورد نظر از عملکرد `lapply()` استفاده می نماید چرا که `y.values` لیستی را ارائه می دهد که می تواند شامل نتایج ارتباطات مختلف در فرایندهای تجربی باشد . ما در این زمینه به پیشرفت هایی دست یافته ایم که در این فصل به آنها اشاره می نماییم . برای هر بردار از مقادیر دقت ، ما دقت تحریف شده را محاسبه نمودیم و از عملکرد `cummax()` و `rev()` استفاده نمودیم . این بخش به سادگی بردار را تغییر می دهد و عملکرد `cummax()` ماکزیمم مجموعه را بدست می آورد . این عملیات ها را تکرار نمائید تا نتیجه مطلوب را کسب کنید . عملکرد `PRcurve()` در بسته ارائه شده ما وجود دارد و شما می توانید از آن استفاده کنید .

در ادامه ما از عملکرد `PRcurve()` برای نمونه اطلاعات ارائه شده استفاده می کنیم تا نمودار سمت راست شکل 4-5 حاصل گردد .

```
> PRcurve(ROCR.simple$predictions, ROCR.simple$labels)
```

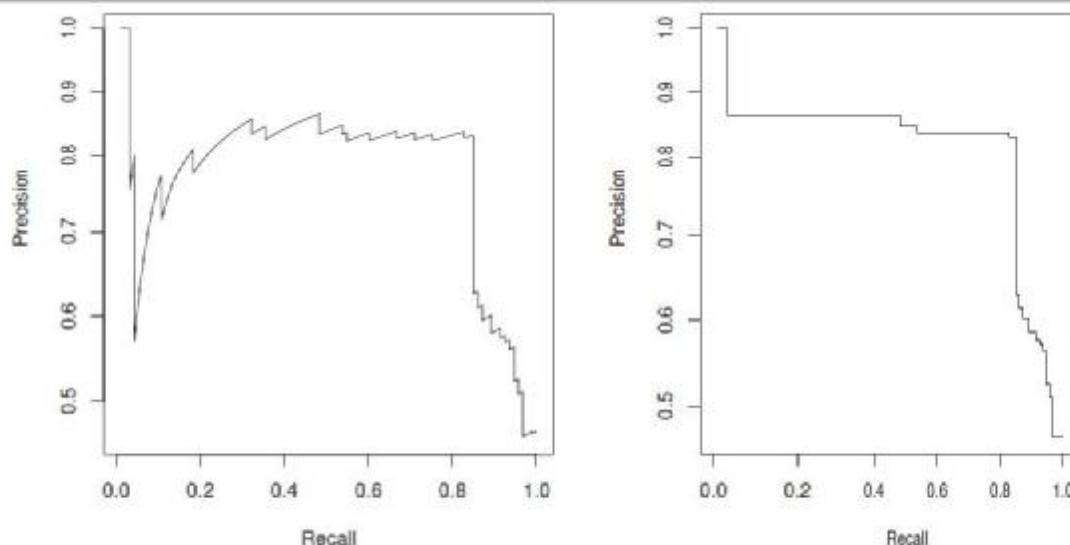


FIGURE 4.5: Smoothed (right) and non-smoothed (left) precision/recall

اما چگونه می توان مدل های رتبه بندی خطاها را ارزیابی نمود به نوع منحنی مربوط می شود . ما در اینجا مجموعه از آزمایش با متغیر `Insp` و مقادیر ممکن `ok`, `fraud`, `unkn` داریم و رتبه بندی مشاهدات در این مجموعه بوسیله همان مدل ایجاد شده است . همچنین نیاز داریم تا امتیاز خطا را برای هر مشاهده مدلسازی نماییم . این امتیاز منبع اطلاعات برای کسب رتبه بندی مشاهدات است .

TABLE 4.1: A Confusion Matrix for the Illustrative Example.

		Predictions		
		ok	fraud	
True Values	ok	3	1	4
	fraud	2	1	3
		5	2	7

اگر مشاهدات مجموعه آزمایش را بوسیله کاهش امتیاز خطا تنظیم نماییم می توانیم مقادیر مختلف دقت و فراخوانی را محاسبه کنیم که به نوع بازبینی و میزان آن بستگی دارد . تنظیم این حد با انتخاب آستانه در امتیاز خطا برابر است که ما مشاهدات را به عنوان موارد دروغین در نظر می گیریم . اجازه دهید تا مثال کوچکی را ارائه دهیم . تصور کنید که ما دارای هفت نمونه آزمایش با مقادیر `ok`, `ok`, `fraud`, `unknown`, `fraud`, `unknown`, `fraud` هستیم که در ستون `Insp` است . مدل مشخصی را در نظر بگیرید که امتیازات خطاها را

برای مشاهدات و مقادیر آن ایجاد می کند: 0.25, 0.3, 0.4, 0.5, 0.7, 0.1, 0.2 که اگر ما مشاهدات را با این امتیازات رتبه بندی کنیم موارد زیر حاصل می شود: fraud, unknown, fraud, fraud, unknown, ok, ok. اگر بازبینی ما تنها به دو مشاهده محدود شود، ارزیابی مدل به صورت دیگری می شود که برای موارد پیش بینی شده به صورت ok, ok, ok, fraud, fraud و برای مقادیر صحیح به صورت ok, ok, unknown, fraud, fraud, unknown است. این مسئله به ماتریکس جدول 4-1 مربوط می شود که در زیر به آن اشاره شده است.

$$Prec = \frac{1}{1+1} = 0.5 \qquad Rec = \frac{1}{2+1} = 0.3333$$

توجه داشته باشید که ما تخمین بدبینانه ای را در زمینه دقت و فراخوانی دنبال نموده ایم که با توجه به گزارشات بازبینی شده می باشد. پیش بینی fraud برای گزارشات موقعیت دوم رتبه بندی، به عنوان خطا در نظر گرفته شده است که ممکن است جعلی یا غیر جعلی باشد. ما از این نوع پردازش برای رتبه بندی خطا استفاده می کنیم تا امتیازات بر اساس منحی PR بدست آید. نمودارهای بالابرنده فراهم کننده دیدگاه مختلف بر اساس پیش بینی های مدل است. این نمودارها اهمیت بیشتری برای مقادیر داشته تا اهداف مناسب حاصل گردد. محور X این نمودارها، مقدار پیش بینی های مثبت (RPP) است که میزان احتمال را نشان می دهد و طبقه مثبت را بررسی می نماید. این میزان بر مقدار کل نمونه های آزمایش تقسیم می شود. در نمونه 1-4 ما مقدار $7/(1+1)$ را دارا هستیم. بر اساس کاربرد نمونه، ما بر اساس بازبینی به آمار توجه نشان می دهیم. محور Y در نمودارهای بالا برنده مقدار فراخوانی تقسیم شده بر مقدار RPP است. نمودارهای بالا برنده با میزان POCR بدست می آید. در زیر به نمودار سمت چپ شکل 4-6 اشاره می نماییم.

```
> pred <- prediction(ROCR.simple$predictions, ROCR.simple$labels)
> perf <- performance(pred, "lift", "rpp")
> plot(perf, main = "Lift Chart")
```

علیرغم نمودارهای بالا برنده ناکارآمد ما دقیقاً" نمی دانیم که در کاربرد خاص چه مطالبی را باید جستجو نماییم .
نموار قابل توجه نشان دهنده مقادیر فراخوانی بر حسب بازنگری است که توسط RPP مشخص می گردد ، ما
این نوع نمودار را نمودار فراخوانی تراکمی می نامیم که می تواند به صورت عملیات زیر اجرا شود .

```
> CRchart <- function(preds, trues, ...) {
+   require(ROCR, quietly = T)
+   pd <- prediction(preds, trues)
+   pf <- performance(pd, "rec", "rpp")
+   plot(pf, ...)
+ }
```

Using again the artificial example, we obtain the right-most graph of Figure 4.6:

```
> CRchart(ROCR.simple$predictions, ROCR.simple$labels,
+   main='Cumulative Recall Chart')
```

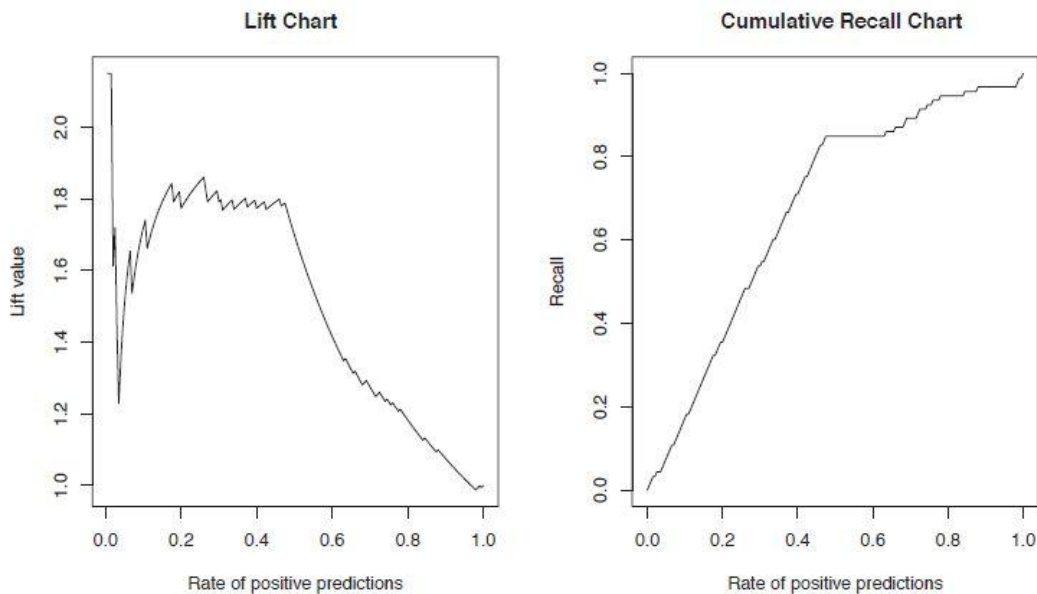


FIGURE 4.6: Lift (left) and cumulative recall (right) charts.

عملکرد `CRchart()` در بسته کتاب ما گنجانده شده است تا در هر زمان بتوان از آن استفاده نمود.

فاصله نرمال شده برای قیمت نمونه :

ارزیابی هایی که در بخش قبلی مورد بررسی قرار گرفت تنها به ارزیابی کیفیت رتبه بندی بر حسب گزارشات مشخص شده می پردازد . این مقادیر در واقع متریک های ارزیابی طبقه بندی است . رتبه بندی ها از طریق مدل هایی بدست می آید که شامل گزارشات غیر مشخص در رتبه بندی است . اما آیا این نمونه ها دقیقاً در رتبه بندی قرار دارند ؟ ما در مورد این مسئله اطمینان کافی نداریم چرا که بازبینی در مورد آنها انجام نداده ایم

. همچنین می توانیم قیمت واحد را با قیمت نمونه گزارشات همان محصول مقایسه نماییم . انتظار می رود که تفاوت بین این قیمت ها بالا باشد که این مسئله نشان دهنده اشکال داشتن گزارش است . فاصله بین قیمت واحد گزارش و قیمت واحد نمونه محصول شاخص خوبی برای کیفیت رتبه بندی حاصله با استفاده از مدل است

محصولات مختلف دارای امتیاز مختلف قیمت واحد هستند همانطور که در شکل 4.4 نشان داده شده است. برای اجتناب از اثرات این تفاوت ها در ارزیابی کیفیت رتبه بندی خطاها باید فاصله بین قیمت واحد نمونه را نرمال سازی نماییم .

$$NDTP_p(u) = \frac{|u - \widetilde{U}_p|}{IQR_p} \quad (4.2)$$

بر اساس آزمایشاتی که انجام شده است می توان از مقدار میانگین NDTPp به عنوان یکی از متریک های ارزیابی استفاده نمود تا عملکرد مدل مشخص گردد . عملکرد زیر به محاسبه مقدار این آمار می پردازد.

```
> avgNDTP <- function(toInsp,train,stats) {
+   if (missing(train) && missing(stats))
+     stop('Provide either the training data or the product stats')
+   if (missing(stats)) {
+     notF <- which(train$Insp != 'fraud')
+     stats <- tapply(train$Uprice[notF],
+                     list(Prod=train$Prod[notF]),
+                     function(x) {
+                       bp <- boxplot.stats(x)$stats
+                       c(median=bp[3],iqr=bp[4]-bp[2])
+                     })
+     stats <- matrix(unlist(stats),
+                     length(stats),2,byrow=T,
+                     dimnames=list(names(stats),c('median','iqr')))
+     stats[which(stats[, 'iqr']==0), 'iqr'] <-
+       stats[which(stats[, 'iqr']==0), 'median']
+   }
+   mdtp <- mean(abs(toInsp$Uprice-stats[toInsp$Prod, 'median']) /
+
+     stats[toInsp$Prod, 'iqr'])
+   return(mdtp)
+ }
```

عملکرد دریافت شده در واقع مجموعه ای از معاملات است که از یک مدل برای بازبینی استفاده می نماید . سپس باید مجموعه ای از موضوعات را دریافت نماید تا مقدار میانگین و IQR هر محصول بدست آید و یا ساختار اطلاعاتی آماده شود که موجب افزایش کارایی محاسباتی عملکرد می گردد . اگر اطلاعات آموزشی دریافت شود ، عملکرد به محاسبه مقادیر میانی و IQR در معاملات غیر جعلی هر محصول می پردازد . در برخی موارد ممکن است IQR صفر باشد به ویژه در مورد محصولاتی که معاملات آنها بسیار کم است . برای اجتناب از صفر شدن محاسبات NDTPp ما مجموعه IQR را برای نمونه های دیگر دارا هستیم .

روش شناسی تجربی :

مجموعه اطلاعات که ما استفاده می کنیم دارای اندازه بسیار معقولی است . در این مقاله از روش Hold Out برای مقایسه های تجربی استفاده شده است . این روش شامل جدا کردن مجموعه اطلاعات در دسترس به صورت رندوم در دو بخش است . یکی از قسمت ها برای بدست آوردن مدل ها استفاده می شود و دیگری برای آزمایش آنها استفاده می ود . این فرایند می تواند به دفعات تکرار شود تا امینان بیشتری حاصل شود . اندازه مجموعه اطلاعات نشان دهنده مقادیری است که ما از لحاظ آماری کسب کرده این . اگر 30% نمونه ها برای آزمایش انتخاب شود به گزارشات 120343 مربوط می شود .

یکی از مشکلات در این بخش ، عدم تعادل بین توزیع نمونه های مختلف گزارشات است . استراتژی نمونه برداری مجدد نرمال ممکن است منجر به مجموعه آزمایش با توزیع مختلف گزارشات طبیعی / جعلی شود . هرگاه ما این نوع توزیع طبقه بندی ناهماهنگ را بدست آوریم ، پیشنهاد می شود که از استراتژی نمونه برداری ساختگی استفاده نماییم . چنین استراتژی شامل نمونه برداری رندوم بر اساس مشاهدات طبقه های مختلف است که این اطمینان را ایجاد می نماید نمونه های انتخاب شده در مجموعه اطلاعات میانی توزیع شده اند . برای نمونه ، اگر ما 10% از نمونه های طبقه X داشته باشیم و باقیمانده یعنی 90 % مربوط به طبقه Y باشد ، آنگاه می توانیم این مشاهدات را در دو بخش جداگانه قرار دهیم . اگر بخواهیم نمونه رندوم ساختگی را با 100 مورد انجام دهیم باید به صورت رندم تعداد 10 نمونه را از طبقه X جدا کرده و تعداد 90 نمونه باقیمانده را از طبقه Y جدا کنیم تا ویژگیهای اصلی طبقه ها حاصل شود .

در بسته کتابی که ما ارائه داده ایم ، عملکرد holdout() وجود دارد که می تواند مورد استفاده قرار گرفته و در آزمایشاتی که دارای عملکرد مشابه هستند دنبال شوند . یکی از پارامترهای این عملکرد ،

طبقه طبقه hldSettings است که مشخص کننده تنظیمات آزمایش است . در میان دیگر پارامترها ، این مورد به شما امکان می دهد تا نمونه های مورد استفاده را مشخص نمایید . در بخش 4-4 چندین مثال در مورد استفاده این نوع عملکرد ارائه شده است تا تخمین های مختلف و آمارهای ارزیابی بررسی شوند . این آمارها از آمارهای دقیق و مربوط به فراخوانی است NDTP است . عملکرد زیر این متریک ها را محاسبه می کند


```

> evalOutlierRanking <- function(testSet,rankOrder,Threshold,statsProds) {
+   ordTS <- testSet[rankOrder,]
+   N <- nrow(testSet)
+   nF <- if (Threshold < 1) as.integer(Threshold*N) else Threshold
+   cm <- table(c(rep('fraud',nF),rep('ok',N-nF)),ordTS$Insp)
+   prec <- cm['fraud','fraud']/sum(cm['fraud',])
+   rec <- cm['fraud','fraud']/sum(cm[, 'fraud'])
+   AVGndtp <- avgNDTP(ordTS[nF,],stats=statsProds)
+   return(c(Precision=prec,Recall=rec,avgNDTP=AVGndtp))
+ }

```

و نیازمند آن است تا مجموعه آزمایش را عرضه نماید و رتبه بندی پیشنهادی را از طریق مدل این مجموعه مشخص نماید که نشان دهنده حد بازبینی و آمارهای محصولات است .

در بخش 2-3-4 ما مشاهده نمودیم که محصولات تفاوت هایی دارند و برخی محصولات با تعداد کمی معاملات در ارتباط هستند . بر این اساس ، پرسش های مختلفی وجود دارد که نیاز به تحلیل معاملات همه محصولات دارد . در این میان متغیرهایی وجود دارد که نیاز به بررسی داشته و برای این کار نیاز به روش ها و تکنیک های مدل سازی است تا بتوان از تغییر پذیری آنها در صورت نیاز بهره برد . اضافه بر آن ، با کنار هم قرار دادن معاملات ، مدل ها بر اساس ارتباطات میان محصول دارای امتیازاتی می شوند . جهت تحلیل چنین نمونه هایی باید معاملات را رتبه بندی نمود و به خطاها امتیاز داد که نیازمند مرحله بالاتر برای کسب رتبه بندی نهایی است . در ادامه رویکردهای مدلسازی را مورد آزمایش قرار می دهیم تا بتوانیم استراتژی مختلفی با توجه به موضوع بدست آوریم . بر اساس روش شناسی تجربی ، ما باید همه محصولات را در کنار هم قرار دهیم تا معاملات مناسب حاصل گردد . این معاملات به طور تصادفی اقدام به انتخاب مجموعه آزمایش می نماید و در این راه از استراتژی hold out استفاده می شود . مجموعه آزمایش با تکنیک های مختلف مدلسازی در ارتباط است و می تواند با رتبه بندی معاملات و بر اساس احتمال جعلی بودن به نتایج خوبی دست یابد . مدل های مورد تحلیل به صورت انفرادی یا جمعی قابل استفاده است .

4-4: بدست آوردن رتبه های خطا

این بخش به شرح مدل های مختلفی می پردازد تا اهداف کسب رتبه ها مشخص شود . برای هر تلاش نتایج بر اساس استراتژی hold out بین 30 تا 70 درصد است . رویکردهای غیر نظارتی :

قانون اصلاح شده Box Plot : در بخش قبلی قانون اصلاح شده Box Plot شرح دادیم که برای شناسایی خطاهای متغیرهای متناوب استفاده می شود . این نمونه در واقع قیمت واحد محصولات است . و ما می توانیم نسبت به این قانون ساده و روش ساختاری آن که برای اطلاعات کاربردی است فکر نماییم . کاربرد قانون اصلاح شده Box Plot برای شناسایی مقادیر غیر معمول قیمت های واحد و معاملات هر محصول می باشد که در تعیین مقادیر به عنوان خطاهای بالقوه نتیجه می دهد . ما می توانیم از این قانون برای هر مجموعه معامله

محصول که در مجموعه آزمایش ارائه شده است استفاده نماییم. در انتها، ما دارای مجموعه ای از خطاهای بالقوه هستیم که برای هر محصول در نظر گرفته شده است. در این راستا باید تصمیم بگیریم که چگونه این مجموعه ها را در رتبه بندی خطا وارد نماییم. این بدان معناست که باید خطاها را به گونه ای مشخص نماییم تا بتوانیم آنها را رتبه بندی کنیم. میزان احتمال استفاده از ایده فاصله نرمال شده برای قیمت واحد نمونه NDTP است در بخش 4.3.2.3 توصیف شده است. این ارزیابی به عنوان قانون Box Plot قابل بررسی است چرا که در هر دو از انواع فاصله با مقادیر مشخص استفاده می شود تا وضعیت مقدار مشخص شود. امتیاز NDTP متریک های بدون واحد است که ما باید آنها را ترکیب نماییم تا مقادیر لازم برای محصولات مختلف بدست آید و بتوانیم برای تمام نمونه های آزمایش رتبه بندی ایجاد نماییم.

```
> BPrule <- function(train,test) {
+   notF <- which(train$Insp != 'fraud')
+   ms <- tapply(train$Uprice[notF],list(Prod=train$Prod[notF]),
+               function(x) {
+                 bp <- boxplot.stats(x)$stats
+                 c(median=bp[3],iqr=bp[4]-bp[2])
+               })
+   ms <- matrix(unlist(ms),length(ms),2,byrow=T,
+               dimnames=list(names(ms),c('median','iqr')))
+   ms[which(ms[, 'iqr']==0), 'iqr'] <- ms[which(ms[, 'iqr']==0), 'median']
+   ORscore <- abs(test$Uprice-ms[test$Prod, 'median']) /
+               ms[test$Prod, 'iqr']
+   return(list(rankOrder=order(ORscore,decreasing=T),
+               rankScore=ORscore))
+ }
```

پارامترهای عملکرد مذکور در واقع مجموعه اطلاعات آموزش است. پس از محاسبه مقادیر IQR در هر محصول می توان عملکرد آمارها را با استفاده از معادله 2-4 بدست آورد. سرانجام لیستی از امتیاز و رتبه بدست می آید. با استفاده از این روش و مقادیر NDTP، کسب امتیاز بر حسب مقادیر متریک امکان پذیر خواهد بود.

در ادامه باید متذکر شد که ما می توانیم از اطلاعات مشابهی بین محصولات استفاده نماییم. در حقیقت محصولاتی که معاملات بسیار کمی دارند قابل بررسی هستند البته اگر محصول بر اساس قیمت واحد توزیع شود. اگر چنین محصولی وجود داشته باشد می توان معاملات آن را اضافه نمود و با استفاده از نمونه بزرگتری که شکل گرفته به ارزیابی پرداخت. این کار بوسیله عملکرد tapply() انجام می شود و اطلاعات محصولات مشابه در فایل SimilarProducts.Rdata ذخیره می شود.

اکنون با استفاده از متدولوژی تجربی به ارزیابی این روش ساده می پردازیم. در ابتدا با محاسبه مقادیر هر محصول که برای امتیاز میانگین NDTP نیاز است کار را شروع می نماییم. همچنین از تمام اطلاعات در دسترس برای این محاسبه استفاده می نماییم تا مقادیر به طور صحیح و با بهترین نتایج به ارزیابی ظرفیت

های رتبه بندی مدل پردازند . به دلیل اینکه اطلاعات جهانی به تکنیک های مدلسازی منتقل نمی شود نمی توان این اطلاعات را از مجموعه آزمایشات برای مدل ها در نظر گرفت . تنها شکل خاصی از تخمین ها برای شناسایی مقادیر غیر معمول بدست می آید .

```
> notF <- which(sales$Insp != 'fraud')
> globalStats <- tapply(sales$Uprice[notF],
+                       list(Prod=sales$Prod[notF]),
+                       function(x) {
+                           bp <- boxplot.stats(x)$stats
+                           c(median=bp[3], iqr=bp[4]-bp[2])
+                       })
> globalStats <- matrix(unlist(globalStats),
+                       length(globalStats), 2, byrow=T,
+                       dimnames=list(names(globalStats), c('median', 'iqr')))
> globalStats[which(globalStats[, 'iqr'] == 0), 'iqr'] <-
+   globalStats[which(globalStats[, 'iqr'] == 0), 'median']
```

عملکرد (`holdout()`) نیاز دارد تا به طور مرتب به ارزیابی روش `BPrule` پردازد که برای هر بخش از فرایند تجربی در نظر گرفته شده است . در بخش های قبلی ما تلاش نمودیم تا عملکردهای مشابهی را که بر اساس کاربر تعریف شده اند جهت دیگر سیستم های یادگیری در آزمایشات تأیید و `Monte Carlo` ایجاد نماییم . این عملکردها از نظر آماری به ارزیابی پرداخته و ارائه دهنده مجموعه آزمایشات و آموزش است . در همین زمان نیاز داریم تا اطلاعات بیشتری کسب نماییم . برای رسم منحنی های فراخوان تراکمی `PR` نیاز است تا عملکردهای `POCR` وجود داشته باشد تا مقادیر صحیح پیش بینی شده بر اساس مشاهدات آزمایش بدست آید . اطلاعات مورد نیاز برای رسم منحنی ها در بخش 2-2-3-4 بیان شده است که به صورت `holdout()` است .

```
> ho.BPrule <- function(form, train, test, ...) {
+   res <- BPrule(train, test)
+   structure(evalOutlierRanking(test, res$rankOrder, ...),
+             itInfo=list(preds=res$rankScore,
+                          trues=ifelse(test$Insp=='fraud', 1, 0)
+             )
+   )
+ }
```

بیشتر اهداف `R` به صورت گسترده توزیع شده اند . در حقیقت دیگر اهداف `R` به بخش پیشین متصل می گردد . معمولاً این اهداف به دنبال کسب اطلاعات بیشتر هستند . در این نمونه ، ما به بردار امتیازات روش `BPrule` نزدیک می شویم که لیستی از مقادیر صحیح و پیش بینی شده از امتیازات اصلی است . عملکرد `structure()` می تواند مورد استفاده قرار گیرد تا مقادیر موضوعات مشخص گردد . این موضوعات باید دارای نام و محتوی باشند . جهت کاربرد ساختاری ما نیاز داریم تا هدفی را بر اساس موضوعات ایجاد نماییم که به آن `itInfo` می

گویند . عملکرد `holdout()` موجب ذخیره کردن این اطلاعات برای هر بخش از آزمایش می شود . همچنین ما نیاز داریم تا هدفی بر اساس موضوع بیان شده ایجاد نماییم . برای ذخیره کردن اطلاعات نیاز داریم تا عملکرد `holdout()` را فراخوانی کنیم که با استفاده از پارامتر بهینه `itsInfo=T` است . این بخش به کاربر اطمینان می دهد تا موضوع تحت عنوان `itInfo` در یک لیست مشخص جمع آوری گردد و نتیجه بر اساس عملکرد `holdout()` بررسی گردد .

اکنون با این عملکرد ما آمادگی داریم تا عملیات `holdout()` را اجرا نماییم و تخمین آمارهای انتخابی را برای سیستم `BPrule` بدست آوریم . بر اساس تنظیمات تجربی ، ما از تقسیمات 30 تا 70 درصدی مجموعه اطلاعات به طور کامل و با استفاده از استراتژی نمونه برداری بهره می گیریم و آمارهای فراخوانی / دقت را برای حد بازبینی از پیش تعیین شده که به میزان 10% است محاسبه می نماییم . این تنظیم آخر قراردادی است و موارد دیگر قابل جایگزینی هستند . عملکرد سیستم دارای حد مختلف بوده و بوسیله منحنی های فراخوانی تراکمی `PR` ارائه خواهد شد . تخمین `hold out` بر اساس تکرار فرایند زیر بدست می آید .

```
> bp.res <- holdOut(learner('ho.BPrule',
+                               pars=list(Threshold=0.1,
+                               statsProds=globalStats)),
+                               dataset(Insp ~ ., sales),
+                               hldSettings(3,0.3,1234,T),
+                               itsInfo=TRUE
+                               )
```

تنظیم پارامتر چهارم از عملکرد `holdOut` () `hldSettings` برای `TRUE` نشان دهنده نمونه برداری طبقه بندی شده است که باید مورد استفاده قرار گیرد . دیگر پارامترها مشخص کننده تعداد تکرار ، درصد نمونه ها و تعداد موارد رندوم است . خلاصه نتایج این آزمایش به صورت زیر بدست می آید .

```
> summary(bp.res)
```

```
== Summary of a Hold Out Experiment ==
```

```
Stratified 3 x 70 %/ 30 % Holdout run with seed = 1234
```

```
* Dataset :: sales
```

```
* Learner :: ho.BPrule with parameters:
```

```
Threshold = 0.1
```

```
statsProds = 11.34 ...
```

```
* Summary of Experiment Results:
```

	Precision	Recall	avgNDTP
avg	0.016630574	0.52293272	1.87123901
std	0.000898367	0.01909992	0.05379945
min	0.015992004	0.51181102	1.80971393
max	0.017657838	0.54498715	1.90944329
invalid	0.000000000	0.00000000	0.00000000

نتایج بررسی های بالا در مورد دقت و فراخوانی بسیار پایین است. معمولاً تنها 52% نمونه های جعلی شناخته شده شامل 10% گزارشات می باشد که بوسیله BPruler ایجاد شده است. مقادیر پایین فراخوانی بدان معناست که 10% تلاش برای همه نمونه های جعلی کافی بوده است اما امکان پذیر نیست که بخشی از نمونه های جعلی در مجموعه آزمایش ارائه شده و میزان دقت پایین باشد. مقدار بسیار پایین دقت به معنای آن است که این روش تنها 10% موقعیت های نمونه های ok, unkn بکار گرفته است. در نمونه گزارشات unkn این مسئله ضرورتاً نامطلوب نیست چرا که ممکن است مربوط به گزارشات جعلی باشد. امتیاز بالای NDTP در واقع مقدار میانگین 1.8 برای NDTP است که به معنای تفاوت بین قیمت واحد این گزارش ها و قیمت میانگین همان محصولات است. میزان IQR شامل 50% گزارشات است که بدین معناست قیمت های واحد این معاملات غیر طبیعی هستند.

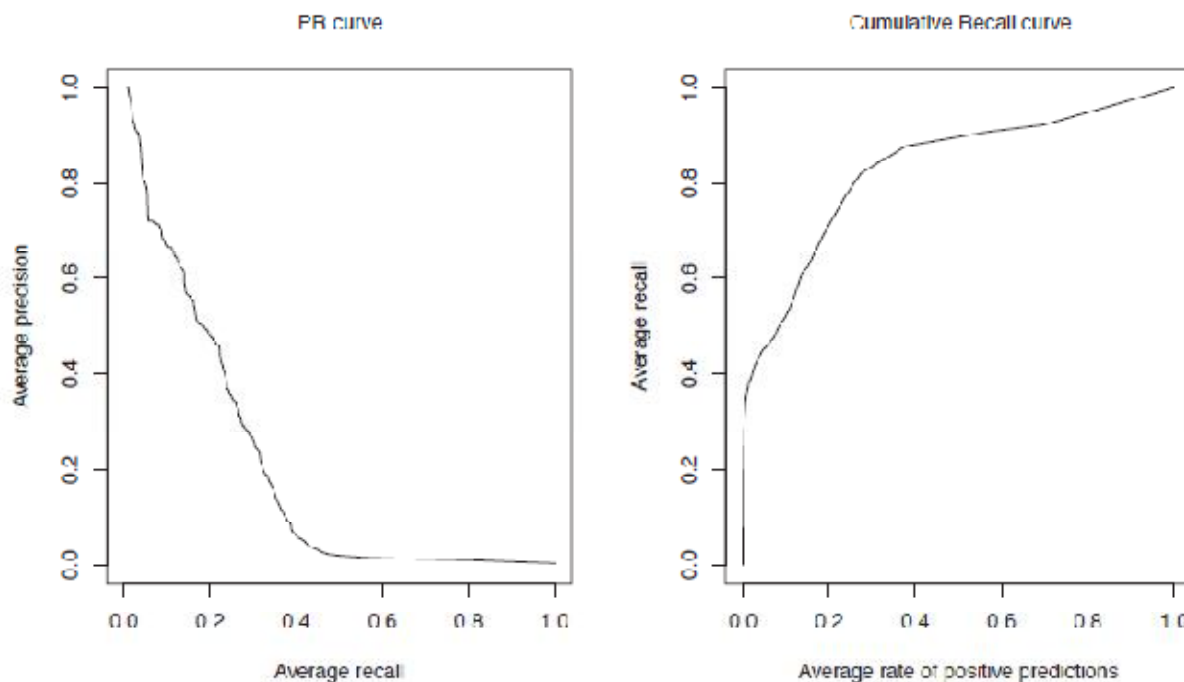
برای بدست آوردن PR و نمودارهای فراخوانی تراکمی، ما نیاز داریم که به امتیازات واقعی خطاها در روش بکار گرفته شده دسترسی داشته باشیم. عملکردی که ما استفاده نموده تا روش رتبه بندی را در هر بخش بکار بندیم به صورت ho.BPrule() است که این مقادیر در بردار آمار توزیع شده است. عملکرد holdout() به جمع آوری اطلاعات اضافه برای هر بخش در لیست می نماید. این لیست با نام itsInfo مشخص است و بوسیله عملکرد holdout() ایجاد شده است. برای کسب اطلاعات لازم در فرمت مذکور باید از عمل کرد ترسیم استفاده کرد و ما نیاز داریم تا گزارشات بیشتری در این زمینه ارائه دهیم. نتیجه کد زیر در منحنی های شکل 4-7 نشان داده شده است.

```

> par(mfrow=c(1,2))
> info <- attr(bp.res,'itsInfo')
> PTs.bp <- aperm(array(unlist(info),dim=c(length(info[[1]]),2,3)),
+                  c(1,3,2)
+                  )
> PRcurve(PTs.bp[, ,1],PTs.bp[, ,2],
+         main='PR curve',avg='vertical')
> CRchart(PTs.bp[, ,1],PTs.bp[, ,2],
+         main='Cumulative Recall curve',avg='vertical')

```

اولین جمله به شما امکان می دهد تا پنجره گراف را به دو بخش تقسیم کنید تا هر دو مورد را در هر بخش تجسم نمایید . جمله دوم از عملکرد `att()` استفاده می نماید تا لیست را استخراج نمایید که شامل مقادیر صحیح و پیش بینی شده `ho.BPrule()` در هر بخش است . این عملکرد را می توان برای بدست آوردن مقدار هر موضوع و تحت نام مشخص استفاده کرد و وارد لیست نمود . سپس این لیست وارد سه بعد می شود . بعد اول نمونه آزمایش است و بعد دوم تکرار تجربه `hold out` است . بعد سوم نوع مقدار را نشان می دهد . برای نمونه ، مقدار `PTs.bp(3,2,1)` مقداری پیش بینی شده از روش مذکور برای آزمایش سوم و تکرار دوم فرایند است . عملکرد `aperm()` برای تغییر ابعاد هر بخش استفاده می شود . اگر متوجه این وضعیت شدید تلاش نمایید تا هر عملکرد را بر اساس بازبینی و نحوه خروجی آن بررسی نمایید .



منحنی های شکل 4-7 بوسیله میانگین مقدار منحنی های هر تکرار در فرایند بدست می آید . نمودار فراخوانی تراکمی عملکرد بهتری را نشان می دهد . ما می توانیم مشاهده کنیم که روش حاصله در حدود 50% از فراخوان

را در بازبینی بسیار پایین بدست آورده است . به هرحال ، برای کسب مقادیر 80 درصدی ، نیاز داریم تا 25 تا 30 درصد گزارشات را مورد بازبینی قرار دهیم .
فاکتورهای خطای داخلی (LOF) :

رتبه بندی خطا ، موضوع مورد تحقیق شناخته شده ای است . در سال 2000 ، Breunig و همکارانش اقدام به توسعه سیستم فاکتورهای خطای داخلی (LOF) نمودند که معمولا "روش رتبه بندی خطا با تکنولوژی جدید است . ایده اصلی این سیستم تلاش برای کسب امتیاز دور از ذهن برای هر نمونه بوسیله تخمین میزان جداسازی با توجه به مجاورت داخلی است . این روش بر اساس میزان مشاهدات در نظر گرفته می شود . نمونه ها در ناحیه با مقدار بسیار پایین به عنوان خطا در نظر گرفته می شوند. با استفاده از فواصل بین نمونه ها تخمین ها حاصل می گردد . ارائه دهندگان این طرح مفاهیم مشخصی را با استفاده از الگوریتم تعریف نمودند تا امتیاز هر بخش را محاسبه کنند . این مفاهیم شامل موارد زیر است :

- ۱- مفهوم فاصله اصلی نقطه p که تعریف کننده فاصله مجاور نزدیکترین k th است .
- ۲- مفهوم فاصله قابل دسترسی بین نمونه $p1$ و $p2$ است که ماکزیمم فاصله اصلی $p1$ و فاصله بین دو نمونه است .
- ۳- فاصله قابل دسترسی داخلی که میزان میانگین قابلیت دسترسی در مجاورهای k را مشخص می کند .

فاکتورهای خطای داخلی (LOF) یک نمونه بوسیله عملکرد فاصله داخلی قابل دسترسی است . بسته کتاب ما شامل تحقق الگوریتم LOF بر اساس تحقیق طراحان مذکور است . در همین زمینه تلاش نمودیم تا عملکرد $lofactor()$ فراهم شود تا مجموعه اطلاعات و اندازه مقادیر LOF مشخص شود . تحقق سیستم LOF به مجموعه اطلاعاتی محدود است که متغیرهای عددی گفته می شود . این بدان معناست که محدودیت متناوب برای الگوریتم های مدلسازی وجود دارد . همانطور که مشاهده نمودیم ، اطلاعات ما شامل متغیرهای عددی مختلف است. این بدان معناست که ما نمی توانیم چنین عملکردی را مستقیما" برای مجموعه اطلاعات خود بکار بندیم . در این زمینه روش های مختلفی برای رویارویی با موضوع وجود دارد . موضوع اول تغییر کد منبع جهت تحقق LOF است که از عملکرد فاصله ترکیبی استفاده میشود.

در این زمینه عملکردهای متعدد فاصله وجود دارد که می تواند مشاهدات بین متغیرهای انواع مختلف را محاسبه می کند . متغیرها شامل کد گذاری مجدد می باشند که به این نوع متغیرها متغیرهای ساختگی گفته می شود و شامل وجود هر مقدار n است . کاربرد این روش در مجموعه اطلاعات ما دارای نوعی مشکل است . متغیرهای ID دارای مقادیر ممکن 6016 است به طوریکه متغیر Prod دارای 4546 است . این بدان معناست که اگر ما از این استراتژی استفاده نماییم می توانیم مجموعه اطلاعات با تعداد 10566 متغیر بدست آوریم . اما این روش نمی تواند برای این مشکل کافی باشد . مورد دیگر شامل ارائه هر محصول به طور جداگانه و بر اساس روش BPrule است . با پردازش این روش ، نه تنها به طور قابل توجهی شرایط محاسباتی کاهش یافته بلکه نیاز برای متغیر Prod رفع شده است . اضافه بر آن ، ارائه محصولات به طور جداگانه همواره رویکردی محتمل در

نظر گرفته می شود و تفاوت هایی بین آنها مشاهده شده است . در غیر اینصورت ، ما باید تصمیم بگیریم که چه اقدامی در مورد اطلاعات فروشندگان ارائه دهیم . رفع این متغیر به معنای این حقیقت است که ما برخی گزارشات غیر واقعی را در مورد فروشندگان در نظر داریم . گرچه این فرضیه خیلی خطرآفرین نمی باشد . حقیقت این است که حتی اگر برخی فروشندگان به کارهای جعلی متهم باشند ممکن است در نحوه گزارش ها یا قیمت واحد اثر نگذارد . پس به طور خلاصه ما از الگوریتم LOF برای مجموعه اطلاعات گزارش ها استفاده می نماییم .

```
> ho.LOF <- function(form, train, test, k, ...) {
+   ntr <- nrow(train)
+   all <- rbind(train, test)
+   N <- nrow(all)
+   ups <- split(all$Uprice, all$Prod)
+   r <- list(length=ups)
+   for(u in seq(along=ups))
+     r[[u]] <- if (NROW(ups[[u]]) > 3)
+       lofactor(ups[[u]], min(k, NROW(ups[[u]]) %% 2))
+       else if (NROW(ups[[u]])) rep(0, NROW(ups[[u]]))
+       else NULL
+   all$lof <- vector(length=N)
+   split(all$lof, all$Prod) <- r
+   all$lof[which(!(is.infinite(all$lof) | is.nan(all$lof)))] <-
+     SoftMax(all$lof[which(!(is.infinite(all$lof) | is.nan(all$lof)))]))
+   structure(evalOutlierRanking(test, order(all[(ntr+1):N, 'lof'],
+                                           decreasing=T), ...),
+             itInfo=list(preds=all[(ntr+1):N, 'lof'],
+                          trues=ifelse(test$Insp=='fraud', 1, 0))
+   )
+ }
```

عملکرد بالا شامل ارزیابی آماری و کاربرد روش LOF برای کسب نتایج است . رویکرد ما بر اساس مجموعه آزمایشات و LOF است تا امتیازات نمونه هایی که به این مجموعه مرتبط است بررسی شود . بر اساس رتبه بندی می توان امتیازات خطاها را انتخاب کرد . همچنین می توان تنها مجموعه آزمایش هایی را رتبه بندی نمود که در آن از اطلاعات استفاده نشده باشد . البته چنین روش غیر نظارتی نمی تواند پیش بینی هایی را به همراه داشته باشد .

عملکرد split() برای تقسیم قیمت های واحد مجموعه اطلاعات محصول بکار می رود . نتیجه در یک لیست ارائه شده است که مولفه های آن قیمت های واحد محصولات هستند . هر loop دارای مجموعه ای از قیمت ها می باشد که در آن از روش LOF استفاده شده تا فاکتور خطا برای هر قیمت حاصل گردد . این فاکتورها در لیست ۲ جمع آوری شده و بوسیله محصول سازماندهی می گردد . اگر حداقل سه گزارش وجود داشته باشد ما تنها از روش LOF استفاده می نماییم وگرنه همه مقادیر نرمال در نظر گرفته می شوند . پس از loop اصلی ،

فاکتورهای خطای حاصله به معاملات متصل می گردد که از عملکرد `split()` در آن استفاده شده است . وضعیت بعدی در واقع تغییر فاکتورهای خطا بر اساس مقیاس 0...1 است . در این بخش از عملکرد `Soft x()` استفاده می شود که با توجه به هدف ما می باشد . این عملکرد میزان زیادی از مقادیر را وارد مقیاس می نماید . با توجه به این حقیقت که عملکرد `lofactor()` مقادیر NaN و Inf را برای معاملات ایجاد می نماید ، ما باید استفاده از عملکرد `Soft x()` را محدود کنیم . سرانجام ، امتیازات ارزیابی رتبه بندی حاصله با مقادیر قابل پیش بینی ، نتیجه این عملکرد است .

مرحله بعد استفاده از فرایند `hold out` جهت کسب تخمین ها در ارزیابی متریک است که بر اساس روش `BPrule` می باشد . ما از همان تنظیمات برای مقادیر تصادفی استفاده نمودیم تا بخش های اطلاعاتی مشخص شوند . بر این اساس مقدار پارامتر `k` در عملکرد `lofactor()` را برای 7 دنبال نمودیم . کلمه اخطار قبل از اجرای کد به سخت افزار شما بستگی دارد که ممکن است مدت زمان طولانی وقت بگیرد و یا حتی تنها چند دقیقه زمان لازم باشد .

```
> lof.res <- holdOut(learner('ho.LOF',
+                               pars=list(k=7,Threshold=0.1,
+                               statsProds=globalStats)),
+                               dataset(Insp ~ .,sales),
+                               hldSettings(3,0.3,1234,T),
+                               itsInfo=TRUE
+                               )
```

```
> summary(lof.res)
```

```
== Summary of a Hold Out Experiment ==
```

```
Stratified 3 x 70 %/ 30 % Holdout run with seed = 1234
```

```
* Dataset :: sales
```

```
* Learner :: ho.LOF with parameters:
```

```
    k = 7
```

```
    Threshold = 0.1
```

```
    statsProds = 11.34 ...
```

```
* Summary of Experiment Results:
```

	Precision	Recall	avgNDTP
avg	0.022127825	0.69595344	2.4631856

std	0.000913681	0.02019331	0.9750265
-----	-------------	------------	-----------

min	0.021405964	0.67454068	1.4420851
-----	-------------	------------	-----------

max	0.023155089	0.71465296	3.3844572
-----	-------------	------------	-----------

invalid	0.000000000	0.000000000	0.000000000
---------	-------------	-------------	-------------

همانطور که مشاهده نمودید ، مقادیر دقت و فراخوانی برای 10% بازبینی بیشتر از مقادیر حاصله بر اساس روش BPrule است . به طور ویژه ، مقدار فراخوانی از 52% به 69% افزایش یافته است . اضافه بر آن ، این امر با افزایش میانگین مقدار NDTP همراه است .

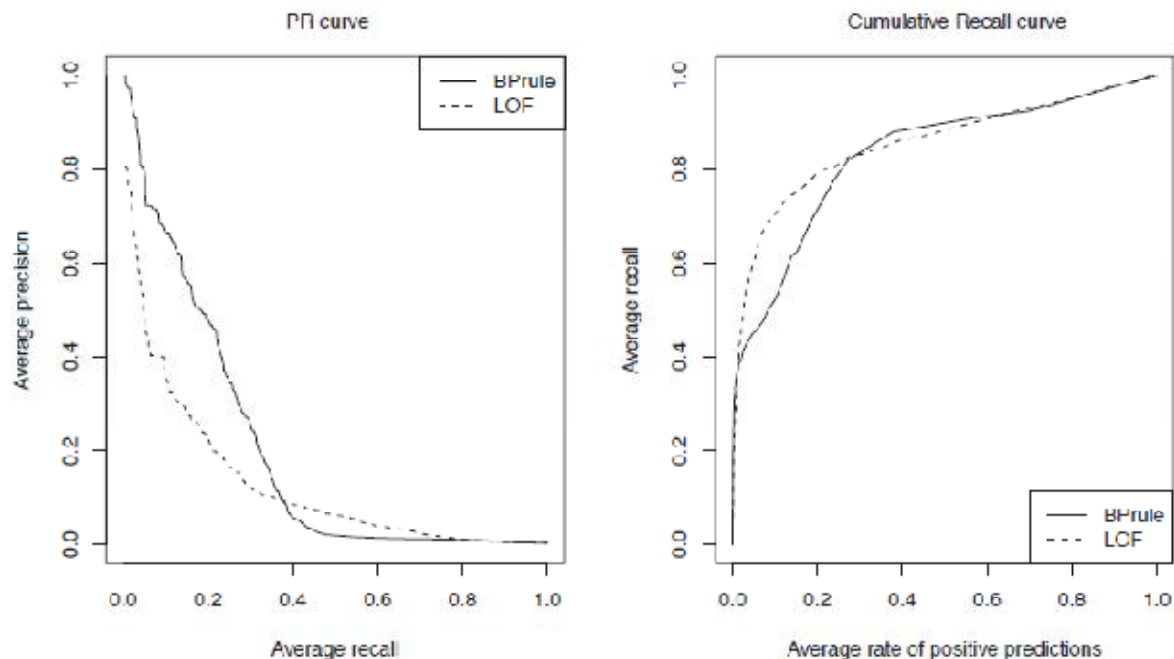
با توجه به منحنی های فراخوانی تراکمی و PR می توان وضعیت بهتری کسب کرد . برای مقایسه بهتر با روش BPrule ، ما منحنی هایی را با این روش رسم نمودیم و از پارامتر add=T استفاده کرده تا بیش از یک منحنی در همان نمودار به نظر برسد .


```

> par(mfrow=c(1,2))
> info <- attr(lof.res,'itsInfo')
> PTs.lof <- aperm(array(unlist(info),dim=c(length(info[[1]]),2,3)),
+                   c(1,3,2)
+                   )
> PRcurve(PTs.bp[, ,1],PTs.bp[, ,2],
+         main='PR curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> PRcurve(PTs.lof[, ,1],PTs.lof[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> legend('topright',c('BPrule','LOF'),lty=c(1,2))
> CRchart(PTs.bp[, ,1],PTs.bp[, ,2],
+         main='Cumulative Recall curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> CRchart(PTs.lof[, ,1],PTs.lof[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> legend('bottomright',c('BPrule','LOF'),lty=c(1,2))

```

تحلیل منحنی های PR نشان می دهد که برای مقادیر فراخوانی ، BPrule می تواند دقت بالاتری را کسب کند . برای مقادیر فراخوانی بالاتر از 40% ، وضعیت فرق دارد . بر حسب فراخوانی که با بازبینی بدست می آید ما می توانیم بگوییم که روش LOF بر BPrule غالب است چرا که بازبینی این روش زیر 25% است . برای مقادیر بالاتر ، تفاوت ها شفاف نیستند و نتایج مقایسه ای می باشند . با توجه به بررسی های انجام شده ، روش LOF جالب تر است چرا که می تواند بین 70 تا 80 درصد خطاها و موارد جعلی را مشخص کند . باید متذکر شد که مقادیر NDTP در روش LOF بوسیله روش BPrule و جهت بازبینی 10 درصدی حاصل شده است .



رتبه بندی خطاها بر اساس دسته بندی (ORh):

روش بعدی، رتبه بندی خطا است که بر اساس الگوریتم دسته بندی شده اند. به این روش، روش رتبه بندی خطاها بر اساس دسته بندی (ORh) می گویند. در این روش از الگوریتم دسته بندی متراکم سلسله مراتبی استفاده می شود تا درخت واره نگار اطلاعات بدست آید. درخت واره نگارها بازنمایی بصری ظهور فرایند روش های دسته بندی است. در پایین ترین سطح ما راه حلی برای گروه های مختلف ارائه

می دهیم. برش این درخت ها در سطوح مختلف ارتفاع موجب ایجاد دسته بندی اطلاعات می گردد. این راه حل میانی از الگوریتم تعاملی بوسیله روش ها استفاده می گردد. مراحل بعدی این الگوریتم تصمیم بر آن دارد که دو گروه مرحله قبلی بتواند در یک دسته واحد ظهور پیدا کند. این فرایند بوسیله معیارهایی هدایت می گردد که تلاش دارد تا مشاهدات مشابه یکدیگر ظاهر یابد. فرایند تعاملی هنگامی توقف می یابد که دو گروه در یک دسته واحد ظاهر شوند. درخت واره نگار به شرح کل فرایند می پردازد. عملکرد `hclust()` در بسته `stats` موجب تحقق متغیرهای مختلف این نوع دسته بندی شود. وضعیت این عملکرد شامل ساختار اطلاعات است که شامل اطلاعاتی در مورد نمونه ها در هر مرحله ظهور است. روش ORh از اطلاعات این ساختار استفاده می نماید تا از روش رتبه بندی خطا استفاده گردد. ایده اصلی این است که خطاها باید مقاومت بیشتری داشته باشند تا بتوانند ظاهر شوند. چنین امری منعکس کننده این ایده است که خطاها باید مشاهدات مختلف داشته باشد. بنابراین، مشاهدات نرمال باید به صورت شفاف بتواند همگنی گروه نتیجه را کاهش دهد. خطاها در مراحل میانی و با دیگر مشاهدات ظاهر می شوند اما در صورتی که خطاها با هم شباهت داشته باشند. این وضعیت در مراحل نهایی فرایند دسته بندی ظهور پیدا می کند و معمولاً "گروه بزرگتر از نمونه ها است."

این ایده کلی توسط روش ORh کسب شده است که امتیاز خطا را برای ه نمونه محاسبه می نماید . برای هر مرحله i دو گروه به صورت مقادیر زیر محاسبه می شوند .

$$of_i(x) = \max \left(0, \frac{|g_{y,i}| - |g_{x,i}|}{|g_{y,i}| + |g_{x,i}|} \right)$$

باید توجه داشت که اعضای گروه بزرگ دارای امتیاز صفر است که در فرایندهای تعاملی ظاهر می گردد . هر مشاهده می تواند به طرق مختلف نمود پیدا کند که فرایند تعاملی آن در واقع الگوریتم دسته بندی سلسله مراتب است . امتیاز نهایی برای هر نمونه اطلاعات ارائه شده است .

عملکرد بسیار مشابه روش LOP است . همانطور که مشاهده نمودیم ، اطلاعات ما شامل متغیرهای عددی مختلف است. این بدان معناست که ما نمی توانیم چنین عملکردی را مستقیماً " برای مجموعه اطلاعات خود بکار بندیم . در این زمینه روش های مختلفی برای رویارویی با موضوع وجود دارد . موضوع اول تغییر کد منبع جهت تحقق LOF است که از عملکرد فاصله ترکیبی استفاده میشود. در این زمینه عملکردهای متعدد فاصله وجود دارد که می تواند مشاهدات بین متغیرهای انواع مختلف را محاسبه می کند .

$$OF_H(x) = \max_i of_i(x)$$

در این بخش از عملکرد outliers.ranking() استفاده می شود که ماتریکس فاصله است و می تواند مشاهدات را رتبه بندی کند .

```

> ho.ORh <- function(form, train, test, ...) {
+   ntr <- nrow(train)
+   all <- rbind(train, test)
+   N <- nrow(all)
+   ups <- split(all$Uprice, all$Prod)
+   r <- list(length=ups)
+   for(u in seq(along=ups))
+     r[[u]] <- if (NROW(ups[[u]]) > 3)
+       outliers.ranking(ups[[u]])$prob.outliers
+       else if (NROW(ups[[u]])) rep(0, NROW(ups[[u]]))
+       else NULL
+   all$orh <- vector(length=N)
+   split(all$orh, all$Prod) <- r
+   all$orh[which(!(is.infinite(all$orh) | is.nan(all$orh)))] <-
+     SoftMax(all$orh[which(!(is.infinite(all$orh) | is.nan(all$orh)))]))
+   structure(evalOutlierRanking(test, order(all[(ntr+1):N, 'orh'],
+                                             decreasing=T), ...),
+             itInfo=list(preds=all[(ntr+1):N, 'orh'],
+                          trues=ifelse(test$Insp=='fraud', 1, 0))
+   )
+ }

```

همانند L OF ، ما توضیحی در مورد مقادیر پارامترهای نداریم و از روش ORh استفاده می کنیم .

```

> orh.res <- holdOut(learner('ho.ORh',
+                               pars=list(Threshold=0.1,
+                                         statsProds=globalStats)),
+                   dataset(Insp ~ ., sales),
+                   hldSettings(3, 0.3, 1234, T),
+                   itsInfo=TRUE
+                   )

```

A summary of the results of the OR_h method is shown below:

```

> summary(orh.res)

```

```

== Summary of a Hold Out Experiment ==

```

```

Stratified 3 x 70 %/ 30 % Holdout run with seed = 1234

```

```

* Dataset :: sales

```

```

* Learner :: ho.ORh with parameters:

```

```

    Threshold = 0.1

```

```

    statsProds = 11.34 ...

```

```

* Summary of Experiment Results:

```

	Precision	Recall	avgNDTP
avg	0.0220445333	0.69345072	0.5444893
std	0.0005545834	0.01187721	0.3712311
min	0.0215725471	0.67979003	0.2893128
max	0.0226553390	0.70133333	0.9703665
invalid	0.0000000000	0.00000000	0.0000000

باید بیان شود که NDTP نتیجه پایین تر از امتیازات دو روش دیگر است . نتایج روش OR_h با LOF قابل مقایسه است و بر اساس منحنی های فراخوان تراکمی این کار انجام می شود . به هر حال ، با توجه به منحنی PR ، سیستم OR_h به روشنی امتیاز LOF را کسب می کند که این امر نشان دهنده امتیاز کمتر BPrule است .

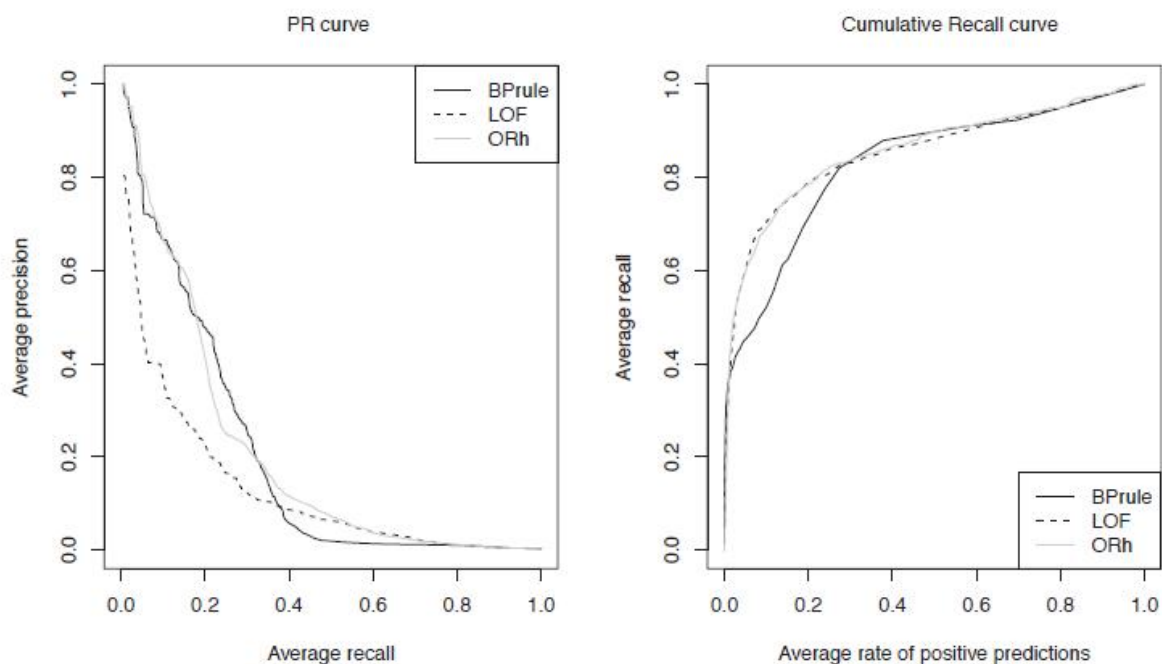

```

> par(mfrow=c(1,2))
> info <- attr(orb.res,'itsInfo')
> PTs.orb <- aperm(array(unlist(info),dim=c(length(info[[1]]),2,3)),
+                   c(1,3,2)
+                   )
> PRcurve(PTs.bp[, ,1],PTs.bp[, ,2],
+         main='PR curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> PRcurve(PTs.lof[, ,1],PTs.lof[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> PRcurve(PTs.orb[, ,1],PTs.orb[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')
> legend('topright',c('BPrule','LOF','ORh'),
+       lty=c(1,2,1),col=c('black','black','grey'))
> CRchart(PTs.bp[, ,1],PTs.bp[, ,2],
+         main='Cumulative Recall curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> CRchart(PTs.lof[, ,1],PTs.lof[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> CRchart(PTs.orb[, ,1],PTs.orb[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')
> legend('bottomright',c('BPrule','LOF','ORh'),
+       lty=c(1,2,1),col=c('black','black','grey'))

```

رویکردهای نظارتی :

در این بخش به بررسی رویکردهای طبقه بندی شده نظارتی اشاره می کنیم که موضوع بحث ما می باشد . هدف ما کسب رتبه بندی برای مجموعه گزارشات معاملات است و باید بر انتخاب مدل ها محدودیت قائل شویم . همچنین از سیستم هایی استفاده می کنیم که قادر باشند طبقه بندی هایی را ایجاد کنند . برای هر نمونه آزمایش ، این روش ها طبقه های مشخصی دارند . این نوع اطلاعات به ما کمک می کند تا گزارشات را بر اساس میزان احتمال رتبه بندی نماییم تا نمونه های جعلی مشخص شوند . قبل از بیان موضوع باید به برخی الگوریتم های طبقه بندی اشاره شود تا موضوع اصلی مجموعه اطلاعات تعیین گردد : توزیع نامتعادل عناوین طبقه ها .



4-4-2- موضوع عدم تعادل طبقه :

مجموعه اطلاعات ما دارای بخش های نامتعادل در گزارشات جعلی و حقیقی است . البته شفافیت نمونه های جعلی در ابهام است و تنها 8.1% گزارشات بازبینی شده وضعیت مشخص دارند . مشکل موجود در دسته بندی مدل ها است . اول اینکه مدل ها نیاز دارند تا ارزیابی دقیقی روی آنها انجام شود تا دقت استاندارد برای آنها تعیین گردد . بر این اساس ، کاربرد ما برای کسب 90% دقت آسان است البته در صورتی که همه گزارشات طبیعی گزارش شوند . از دیگر موضوعات عدم تعادل طبقه و تاثیر قوی آن بر عملکرد الگوریتم های یادگیری است که اقلیت طبقه ها دارای حمایت آماری نیستند . این وضعیت مشکل ساز در برخی نمونه ها وضعیتی غالب پیدا می کند .

در همین ارتباط تکنیک های مختلفی وجود دارد که با هدف کمک به غلبه بر مشکلات الگوریتم های یادگیری وارد شده و بوسیله بررسی عدم تعادل این موضوع مشخص می گردد . در این ارتباط دو گروه عمده وجود دارد :

۱- روش هایی که فرایند یادگیری را با استفاده از متریک ارزیابی خاص دنبال می کنند و به نمونه های اقلیت حساس هستند

۲- روش های نمونه برداری که اطلاعات آموزشی را دنبال می نماید تا توزیع تغییر کند .

تلاش ما استفاده از روش های طبقه بندی نظارتی می باشد که در این مسیر از روش دوم استفاده شده است . روش های نمونه برداری مختلفی پیشنهاد شده است تا عدم تعادل اطلاعات تغییر کند . تحت روش های فوق ، بخش کوچکی از اکثریت نمونه های طبقات انتخاب و به حداقل نمونه ها اضافه می شود تا مجموعه اطلاعات با

توزیع متعادل حاصل گردد . روش های نمونه برداری با استفاده از فرایند جایگزینی به این کار ادامه می دهد . یکی از نمونه های مناسب ، روش SMOTE است ایده کلی این روش با استفاده از موضوعات مجاور نمونه ها صورت می گیرد . اضافه بر آن ، اکثریت نمونه ها تحت مجموعه اطلاعات متعادل هستند . ما این نوع نمونه برداری را طی عملکردی محقق ساختیم که SMOTE() نامیده و در کتاب خود ارائه داده ایم . با ارائه نمونه های غیر متعادل ، این عملکرد می تواند مجموعه اطلاعات جدیدی را ایجاد کند . کد زیر نشان دهنده این عملکرد است .

```
> data(iris)
> data <- iris[, c(1, 2, 5)]
> data$Species <- factor(ifelse(data$Species == "setosa", "rare",
+                               "common"))
> newData <- SMOTE(Species ~ ., data, perc.over = 600)
> table(newData$Species)
```

```
common  rare
    600    350
```

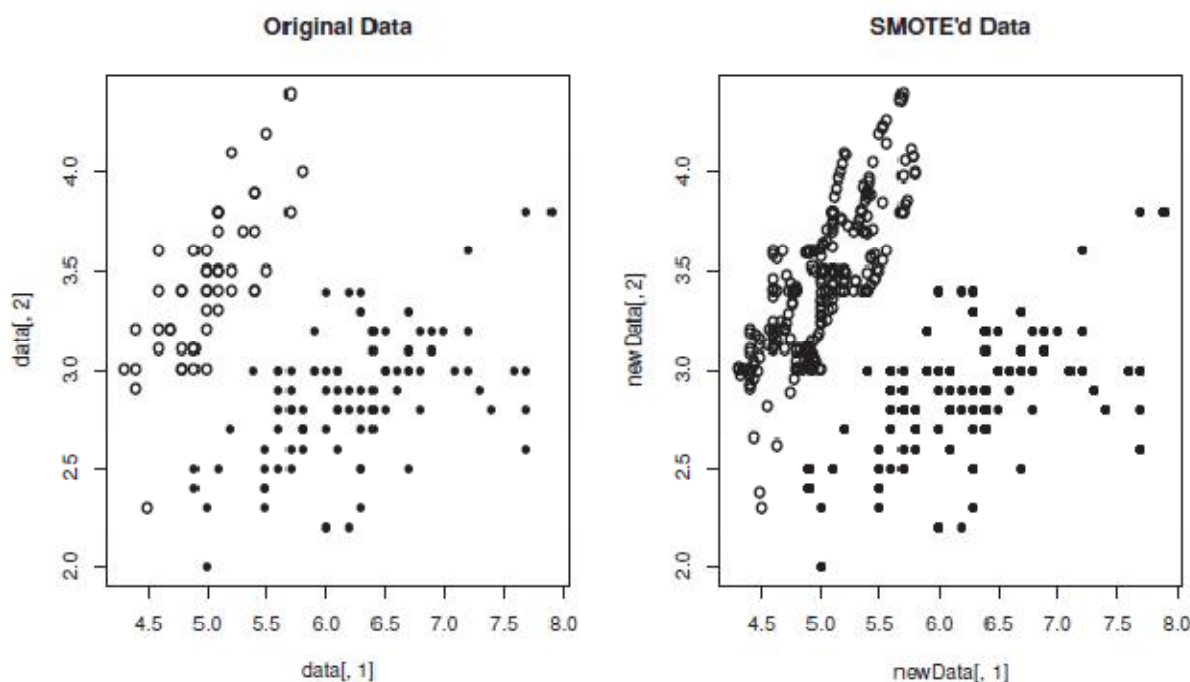
در

دیگر مثال از اطلاعات iris استفاده شده است که می تواند مجموعه اطلاعات ساختگی را با دو متغیر پیش بینی شده ایجاد نماید و متغیر جدیدی ارائه دهد . عملکرد این کد به صورت SMOTE() است و که با مقدار 600 و پارامتر perc.over انجام می گیرد که بدان معناست شش نمونه جدید برای هر بخش ایجاد شده است که در مجموعه اطلاعات قرار دارد . نمونه های جدید به صورت رندوم بین نمونه های مجاور قرار می گیرند . با استفاده از SMOTE() می توانید مجموعه اطلاعاتی با متغیرهای متناوب و عددی داشته باشید .

```
> par(mfrow = c(1, 2))
> plot(data[, 1], data[, 2], pch = 19 + as.integer(data[, 3]),
+       main = "Original Data")
> plot(newData[, 1], newData[, 2], pch = 19 + as.integer(newData[,
+       3]), main = "SMOTE'd Data")
```

در ادامه ایده بهتری برای آنچه که در ایجاد نقشه موثر است شکل داده ایم که بر اساس مجموعه اطلاعات SMOTE'S است . نتایج در ذیل نشان داده شده است . در تمامی آزمایشات ما از الگوریتم طبقه بندی نظارتی استفاده نمودیم و تلاش بر آن داریم که با استفاده از مجموعه های آموزشی متعادل و روش SMOTE متغیرهای دیگری ایجاد نماییم .

شکل 4-10: استفاده از SMOTE جهت ایجاد نمونه های کمیاب بیشتر



4-4-2-2 : Naïve Bayes

Naïve Bayes یک طبقه بندی کننده احتمالاتی بر اساسی نظریه Bayes است که از فرضیات بسیار قوی بین پیش بینی ها استفاده می نماید . این فرضیات کمتر به نمونه های دنیای واقعی مربوط است و به همین دلیل نام naive روی آن گذاشته شده است . این روش به طور موفقیت آمیز در نمونه های وسیع و کاربردهای دنیای واقعی استفاده می شود . طبقه بندی کننده Naïve Bayes به محاسبه احتمالات در هر طبقه و برای هر نمونه می پردازد به طوریکه اگر C طبقه و X_1, \dots, X_p مقادیر مشاهده شده پیش بینی ها برای نمونه باشد به صورت زیر محاسبه می شود .

$$P(c|X_1, \dots, X_p) = \frac{P(c)P(X_1, \dots, X_p|c)}{P(X_1, \dots, X_p)}$$

احتمال $P(c)$ به عنوان مقدار امید در طبقه C در نظر گرفته می شود . $P(X_1, \dots, X_p|c)$ در واقع میزان احتمال نمونه آزمایش طبقه C است . سرانجام ، مخرج مقدار احتمال مشاهده شده است . معادله مذکور برای همه مقادیر ممکن در نظر گرفته شده است تا بیشترین میزان احتمال تعیین گردد . این وضعیت به نمونه عددی معادله مربوط می گردد . با استفاده از تعاریف آماری در زمینه احتمالات شرایطی ، باید مقدار وابستگی بین پیش بینی ها مشخص شود تا بتوانیم صورت کسر معادله را کاهش دهیم .

پیاده سازی روش Naïve Bayes می تواند میزان احتمالات را از نمونه آموزشی و با استفاده از فرکانس های نسبی تخمین بزند . با استفاده از این تخمین ها ، روش اقدام به خروج برخی احتمالات برای هر نمونه آزمایش بر اساس معادله 4-5 می نماید .

$$P(c)P(X_1, \dots, X_p|c) = P(c) \prod_{i=1}^P P(X_i|c)$$

R دارای چندین پیاده سازی بر اساس روش Naïve Bayes است . ما می توانیم از عملکرد naiveBayes() از e1071 استفاده نماییم . بسته klaR شامل پیاده سازی این طبقه کننده ها است که موجب افزایش عملکرد می گردد .

```
> nb <- function(train, test) {
+   require(e1071, quietly = T)
+   sup <- which(train$Insp != "unkn")
+   data <- train[sup, c("ID", "Prod", "Uprice", "Insp")]
+   data$Insp <- factor(data$Insp, levels = c("ok", "fraud"))
+   model <- naiveBayes(Insp ~ ., data)
+   preds <- predict(model, test[, c("ID", "Prod", "Uprice",
+     "Insp")], type = "raw")
+   return(list(rankOrder = order(preds[, "fraud"], decreasing = T),
+     rankScore = preds[, "fraud"]))
+ }
```

عملکردهای زیر از Naïve Bayes استفاده می کنند که برای کسب امتیازات رتبه بندی در مجموعه آزمایش گزارشات در نظر گرفته می شود . همچنین از گزارشات بازبینی شده از نمونه آموزشی استفاده می شود تا مدل Naïve Bayes بدست آید . رتبه بندی خطا با استفاده از احتمالات تخمین زده شده بدست می آید .

```
> ho.nb <- function(form, train, test, ...) {
+   res <- nb(train, test)
+   structure(evalOutlierRanking(test, res$rankOrder, ...),
+     itInfo=list(preds=res$rankScore,
+       trues=ifelse(test$Insp=='fraud', 1, 0)
+     )
+   )
+ }
```

سرانجام از عملکرد holdout() استفاده می شود تا آزمایشات به همراه تنظیمات استفاده شود و مدل های غیر نظارتی در بخش های قبل مورد توجه قرار بگیرند . نتایج مدل Naïve Bayes برای 10% فعالیت بازبینی است .

```
> nb.res <- holdOut(learner('ho.nb',
+                               pars=list(Threshold=0.1,
+                                         statsProds=globalStats)),
+                   dataset(Insp ~ ., sales),
+                   hldSettings(3, 0.3, 1234, T),
+                   itsInfo=TRUE
+                   )
```

نتایج مدل بالا برای 10% بازبینی به صورت زیر است :

```
> summary(nb.res)
```

```
== Summary of a Hold Out Experiment ==
```

```
Stratified 3 x 70 %/ 30 % Holdout run with seed = 1234
```

```
* Dataset :: sales
```

```
* Learner :: ho.nb with parameters:
```

```
Threshold = 0.1
```

```
statsProds = 11.34 ...
```

```
* Summary of Experiment Results:
```

	Precision	Recall	avgNDTP
avg	0.013715365	0.43112103	0.8519657
std	0.001083859	0.02613164	0.2406771
min	0.012660336	0.40533333	0.5908980
max	0.014825920	0.45758355	1.0650114
invalid	0.000000000	0.00000000	0.0000000

امتیازات به طور قابل توجهی روبه نقصان می روند و بهترین نتایج با روش غیر نظارتی حاصل می گردد . در ادامه به منحنی ها توجه می کنیم که عملکرد بهتری را نشان می دهند سپس به مقایسه Naïve Bayes با یکی از مدل های غیر نظارتی می پردازیم .

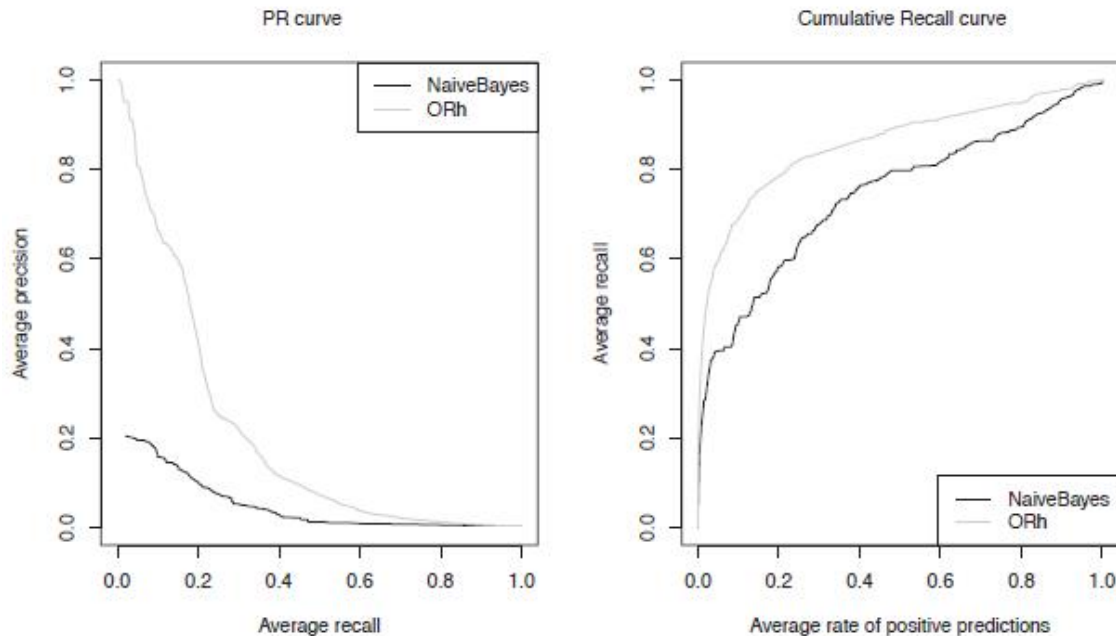
```

> par(mfrow=c(1,2))
> info <- attr(nb.res,'itsInfo')
> PTs.nb <- aperm(array(unlist(info),dim=c(length(info[[1]]),2,3)),
+                  c(1,3,2)
+                  )

> PRcurve(PTs.nb[, ,1],PTs.nb[, ,2],
+         main='PR curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> PRcurve(PTs.orh[, ,1],PTs.orh[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')
> legend('topright',c('NaiveBayes','ORh'),
+        lty=1,col=c('black','grey'))
> CRchart(PTs.nb[, ,1],PTs.nb[, ,2],
+         main='Cumulative Recall curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> CRchart(PTs.orh[, ,1],PTs.orh[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')
> legend('bottomright',c('NaiveBayes','ORh'),
+        lty=1,col=c('black','grey'))

```

نمودارهای 4-11 شفافیت بالایی دارد و بر اساس Naïve Bayes می باشد که روش ORh را برای کاربردهای خاص در نظر دارد . هر دو منحنی نشان می دهد که روش بعدی بر تمام تنظیمات غالب است. در صورت عملکرد ضعیف Naïve Bayes ممکن است عدم تعادل این موضوع به وضوح مشخص گردد . در بخش 1-2-4-4 به بررسی روش هایی برای بیان این موضوع پرداخته و با استفاده از الگوریتم SMOTE کار را دنبال می کنیم . اکنون از طبقه بندی کننده Naïve Bayes استفاده کرده و با روش SMOTE کار را دنبال می کنیم .



شکل 4-11: منحنی های فراخوانی (سمت راست) و PR (سمت چپ) در روش های ORh و Naïve Bayes

تفاوت اصلی کد قبلی به عملکرد زیر مربوط می شود جاییکه عملکرد `naiveBayes()` فراخوانده شده و نیاز به اصلاح وجود دارد.

```
> nb.s <- function(train, test) {
+   require(e1071, quietly = T)
+   sup <- which(train$Insp != "unkn")
+   data <- train[sup, c("ID", "Prod", "Uprice", "Insp")]
+   data$Insp <- factor(data$Insp, levels = c("ok", "fraud"))
+   newData <- SMOTE(Insp ~ ., data, perc.over = 700)
+   model <- naiveBayes(Insp ~ ., newData)
+   preds <- predict(model, test[, c("ID", "Prod", "Uprice",
+   "Insp")], type = "raw")
+   return(list(rankOrder = order(preds[, "fraud"], decreasing = T),
+   rankScore = preds[, "fraud"]))
+ }
```

وضعیت های زیر برای نسخه SMOTE از Naïve Bayes در نظر گرفته شده است:


```

> ho.nbs <- function(form, train, test, ...) {
+   res <- nb.s(train, test)
+   structure(evalOutlierRanking(test, res$rankOrder, ...),
+             itInfo=list(preds=res$rankScore,
+                          trues=ifelse(test$Insp=='fraud', 1, 0)
+             )
+   )
+ }

> nbs.res <- holdOut(learner('ho.nbs',
+                             pars=list(Threshold=0.1,
+                                         statsProds=globalStats)),
+                   dataset(Insp ~ ., sales),
+                   hldSettings(3, 0.3, 1234, T),
+                   itsInfo=TRUE
+                   )

```

نتایج

این بخش از مدل Naïve Bayes تنها برای 10% بازبینی مناسب است .

```
> summary(nbs.res)
```

```
== Summary of a Hold Out Experiment ==
```

```
Stratified 3 x 70 %/ 30 % Holdout run with seed = 1234
```

```
* Dataset :: sales
```

```
* Learner :: ho.nbs with parameters:
```

```
Threshold = 0.1
```

```
statsProds = 11.34 ...
```

```
* Summary of Experiment Results:
```

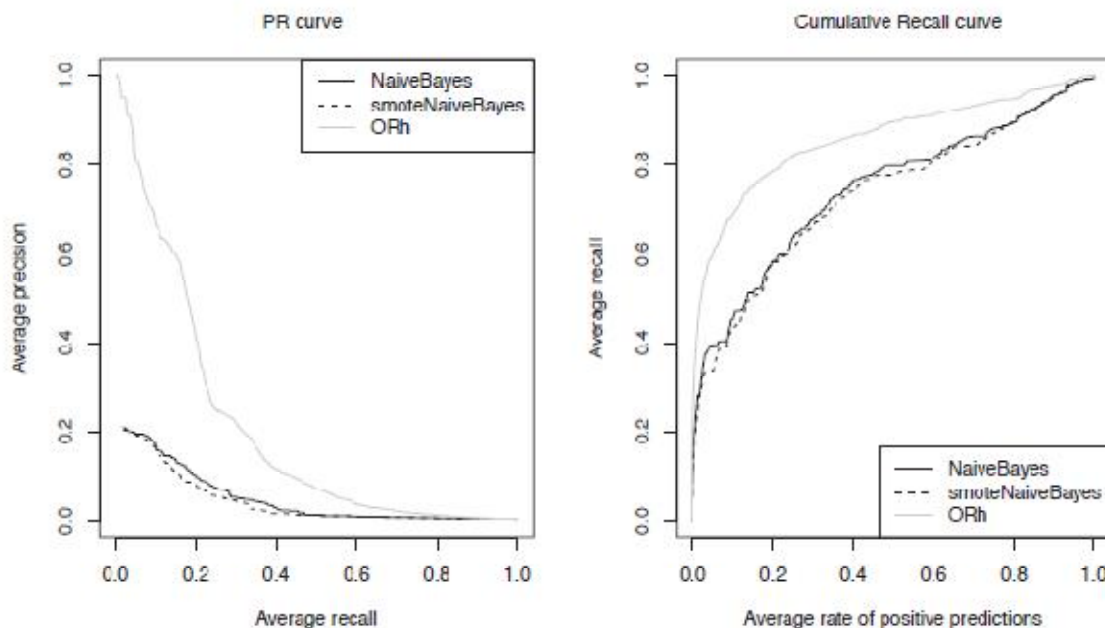
	Precision	Recall	avgNDTP
avg	0.014215115	0.44686510	0.8913330
std	0.001109167	0.02710388	0.8482740
min	0.013493253	0.43044619	0.1934613
max	0.015492254	0.47814910	1.8354999
invalid	0.000000000	0.000000000	0.0000000

این نتایج با توجه به نتایج طبیعی Naïve Bayes غیر قابل تشخیص است . امتیازات تنها می توانند نوعی برتری را نشان دهند اما بهترین نتایج از مدل های غیر نظارتی حاصل شده است . به نظر می رسد علیرغم نمونه

بردارى بیش از حد از موضوعات که توسط روس SMOTE صورت گرفته است باز هم روش Naïve Bayes قادر به پیش بینی صحیح نمی باشد و گزارشات جعلی به وفور به چشم می خورد .

```
> par(mfrow=c(1,2))
> info <- attr(nbs.res,'itsInfo')
> PTs.nbs <- aperm(array(unlist(info),dim=c(length(info[[1]]),2,3)),
+                  c(1,3,2)
+                  )
> PRcurve(PTs.nb[, ,1],PTs.nb[, ,2],
+         main='PR curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> PRcurve(PTs.nbs[, ,1],PTs.nbs[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> PRcurve(PTs.orh[, ,1],PTs.orh[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')
> legend('topright',c('NaiveBayes','smoteNaiveBayes','ORh'),
+        lty=c(1,2,1),col=c('black','black','grey'))
> CRchart(PTs.nb[, ,1],PTs.nb[, ,2],
+         main='Cumulative Recall curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> CRchart(PTs.nbs[, ,1],PTs.nbs[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> CRchart(PTs.orh[, ,1],PTs.orh[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')
> legend('bottomright',c('NaiveBayes','smoteNaiveBayes','ORh'),
+        lty=c(1,2,1),col=c('black','black','grey'))
```

نمودارهای شکل 12 تأیید کننده نتایج نامطلوب نسخه SMOTE از Naïve Bayes است . در حقیقت ، این نمودارها نشان دهنده همان نتایج ضعیف Naïve Bayes استاندارد است که با ORh مقایسه شده است . اضافه بر آن ، این عملکرد همواره بهتر از نسخه استاندارد عمل نموده است .



شکل 4-12: منحنی های فراخوانی (سمت راست) و PR (سمت چپ) برای دو بخش دیگر روش های ORh و Naïve Bayes

با توجه به نتایج عرضه شده ، ما باید این پرسش را مطرح کنیم که آیا ما محدودیت های مدل توسط محصول را به تصویر کشیده ایم و یا تنها به روش های غیر نظارتی بسنده نموده ایم که ممکن است موجب ایجاد مشکلات با مدل شود . بر این اساس نیاز داریم تا رویکرد جدیدی را با Naïve Bayes شروع نماییم . همچنین نیاز است تا کد مورد استفاده برای مدل های غیر نظارتی وفق داده شود تا معادلات را بر اساس روش Naïve Bayes به تصویر بکشد . مشکل اضافی در این بخش به کمبود گزارشات نظارتی مربوط می شود . در حقیقت با وجود محدودیت ما شاهد محصولاتی هستیم که دارای معاملات متعدد هستند و این امر مشکوک می باشد . اگر محدودیت را تنها با استفاده از معادلات مشخص ایجاد نماییم ممکن است بتوانیم این مشکل را رفع کنیم .

3-2-4-4 : AdaBoost

AdaBoost نوعی الگوریتم یادگیری برای طبقه ای از روش های دسته جمعی است . این مدل ها بوسیله مجموعه ای از بخش ها شکل پیدا کرده که برای شناسایی الگوریتم و با استفاده از نوعی تراکم توزیع می گردند . AdaBoost از روش وفق پذیری افزایشی استفاده نموده تا مجموعه ای از مدل های پایه را ایجاد نماید . مدل افزایشی در واقع روش کلی است که می تواند برای بهبود عملکرد الگوریتم مورد استفاده قرار گیرد و بهتر از طبقه بندی کننده رندوم عمل می نماید . ایجاد مدل AdaBoost به صورت متوالی حاصل می شود . هر عضو جدید در توالی از طریق بهبود خطاهای مدل قبلی ظاهر می شود . بهبودی ها از طریق طرح توزین بدست آمده و موجب افزایش وزن نمونه ها می گردد که بوسیله مدل قبلی طبقه بندی می شود . این بدان معناست که

فراگیرنده اصلی در توزیع مختلف اطلاعات آموزشی مورد استفاده قرار میگیرد . پس از تعاملات متعدد در این فرایند ، نتیجه در واقع مجموعه ای از مدل های حاصله در نمونه های مختلف است . چنین وضعیتی برای کسب پیش بینی ها مناسب است و نمونه های آزمایش به صورت اصلی باید مورد توجه قرار گیرد . پیش بینی ها بوسیله میانگین توزین شده در نظر گرفته می شوند و نیازمند مدل های پایه هستند . این توزین ها تعریف شده و مقادیر بزرگتری در توالی به خود کسب کرده اند .

طرح نمونه توزین که بوسیله روش AdaBoost مورد استفاده قرار می گیرد بر اساس توزیع عدم تعادل قابل توجه بوده و دیدگاه یادگیری را نشان می دهد . حتی اگر در بخش های داخلی نمونه هایی از طبقه اقلیت وجود داشته باشد که بوسیله مدل ها تعیین شده باشند ، وزن آنها افزایش یافته و مدل نیروی مجدد می گیرد که روی یادگیری تاکید دارد . چنین وضعیتی منجر به ایجاد دقت بیشتر در پیش بینی نمونه های کمیاب می شود .

AdaBoost.M1 یکی از بخش های خاص روش AdaBoost می باشد و در آن یادگیرنده اصلی به طبقه بندی ارقام می پردازد . این روش در عملکرد `adaboost.M1()` در بسته `adabag()` تحقق می یابد . متأسفانه ، روش `predict` برای مدل هایی ایجاد شده است که توانایی برگشتن احتمالات طبقه را ندارند . این مسئله محدودیتی پیچیده برای فعالیت ما محسوب می شود . همانطور که قبلاً بیان شد ، ما نیاز داریم تا احتمالات طبقه داشته باشیم چرا که در هر گزارش از آنها استفاده کرده و موارد دروغین را کشف کنیم و رتبه بندی خطا بدست آوریم . در ادامه ما از پیاده سازی الگوریتم AdaBoost.M1 استفاده نمی کنیم . از طرفی دیگر ، در کتابی که پیش روی شماست از چنین پیاده سازی استفاده شده است . . به هر حال ، از نرم افزار استخراج اطلاعات Weka نیز استفاده می نماییم . این نرم افزار منبع باز و مناسبی برای استخراج اطلاعات و یادگیری ماشینی است . در واقع این نرم افزار ابزار مناسبی محسوب می شود که الگوریتم های یادگیری را با نمونه های مناسب گرافیکی ارائه می دهد . در مقایسه با R ف این نرم افزار بهتر عمل نموده است چرا که رابط ساده و مناسبی با کاربر است گرچه R انعطاف پذیری بیشتر داشته و بر حسب توسعه نرم افزار و ابزار مدلسازی و ایجاد نمودار گسترده تر عمل نموده است . نصب این بسته نرم افزاری به شما کمک می کند تا برنامه های مختلفی داشته باشید البته نیاز است که قبلاً "برنامه جاوا را نصب کرده باشید . فرایند نصب موجب می شود تا دستورالعمل های شفافی برای نمونه های درخواستی شما فراهم گردد . ما به شما پیشنهاد می کنیم که این بسته نرم افزاری را در کامپیوتر خود نصب کرده و با کمک آن روش های متعدد را بیابید و از RWeka بهره بگیرید .

عملکرد `AdaBoostM1()` در بسته RWeka موجود است و امکان طبقه بندی نمونه های AdaBoost.M1 با استفاده از پیاده سازی Weka و این الگوریتم وجود دارد . در مقایسه با پیاده سازی بسته `adabag` ، روش `predict` از این الگوریتم می تواند طبقه بندی خوبی ارائه دهد و با استفاده از آن رتبه بندی خطا بهتر صورت گیرد . به طور پیش فرض ، پیاده سازی Weka از تصمیماتی استفاده می نماید . این مدل ها در واقع نوع خاصی محسوب می شوند و تنظیم آنها نیازمند پارامترهایی از عملکرد است تا تغییرات صورت گیرد . عملکرد

WOW() به شما این امکان را می دهد تا پارامترهای در دسترس را برای الگوریتم یادگیری Weka بررسی نمایید .

- P Percentage of weight mass to base training on. (default 100, reduce to around 90 speed up)
Number of arguments: 1.
- Q Use resampling for boosting.
- S Random number seed. (default 1)
Number of arguments: 1.
- I Number of iterations. (default 10)
Number of arguments: 1.
- D If set, classifier is run in debug mode and may output additional info to the console
- W Full name of base classifier. (default: weka.classifiers.trees.DecisionStump)
Number of arguments: 1.
-
- D If set, classifier is run in debug mode and may output additional info to the console

مقدار هر پارامتر قابل تغییر است و شما می توانید با عملیات بازبینی آن ها را فراخوان نمایید . استفاده از عملیات Weka – control() برای این بخش لازم است . در اینجا نمونه ای تشریحی از کاربرد AdaBoostM1() ارائه می دهیم .

```

preds      setosa versicolor virginica
setosa      19         0         0
versicolor   0        13         1
virginica    0         2        15

```

```

> prob.preds <- predict(model,iris[-idx,],type='probability')
> head(prob.preds)

```

```

      setosa  versicolor  virginica
2 0.9999942 5.846673e-06 2.378153e-11
4 0.9999942 5.846673e-06 2.378153e-11
7 0.9999942 5.846673e-06 2.378153e-11
9 0.9999942 5.846673e-06 2.378153e-11
10 0.9999942 5.846673e-06 2.378153e-11
12 0.9999942 5.846673e-06 2.378153e-11

```

```

> data(iris)
> idx <- sample(150,100)
> model <- AdaBoostM1(Species ~ .,iris[idx,],
+                      control=Weka_control(I=100))
> preds <- predict(model,iris[-idx,])
> head(preds)

```

```

[1] setosa setosa setosa setosa setosa setosa
Levels: setosa versicolor virginica

```

```

> table(preds,iris[-idx,'Species'])

```

مثال بالا مشخص کننده چگونگی طبقه بندی احتمالی با استفاده از این مدل است.

```

> ab <- function(train,test) {
+   require(RWeka,quietly=T)
+   sup <- which(train$Insp != 'unkn')
+   data <- train[sup,c('ID','Prod','Uprice','Insp')]
+   data$Insp <- factor(data$Insp,levels=c('ok','fraud'))
+   model <- AdaBoostM1(Insp ~ .,data,
+                       control=Weka_control(I=100))
+   preds <- predict(model,test[,c('ID','Prod','Uprice','Insp')],
+                   type='probability')
+   return(list(rankOrder=order(preds[, 'fraud'],decreasing=T),
+              rankScore=preds[, 'fraud']))
+ }

```

سرانجام ما دارای کدهای زیر برای اجرای آزمایشات hold out هستیم :

```
> ab.res <- holdOut(learner('ho.ab',  
+                        pars=list(Threshold=0.1,  
+                        statsProds=globalStats)),  
+                  dataset(Insp ~ .,sales),  
+                  hldSettings(3,0.3,1234,T),  
+                  itsInfo=TRUE  
+                  )
```

نتایج AdaBoost برای 10% از تلاش به صورت زیر است :

```
> summary(ab.res)
```

```
== Summary of a Hold Out Experiment ==
```

```
Stratified 3 x 70 %/ 30 % Holdout run with seed = 1234
```

```
* Dataset :: sales
```

```
* Learner :: ho.ab with parameters:
```

```
Threshold = 0.1
```

```
statsProds = 11.34 ...
```

```
* Summary of Experiment Results:
```

	Precision	Recall	avgNDTP
avg	0.0220722972	0.69416565	1.5182034
std	0.0008695907	0.01576555	0.5238575
min	0.0214892554	0.68241470	0.9285285
max	0.0230717974	0.71208226	1.9298286
invalid	0.0000000000	0.00000000	0.0000000

اکنون عملکرد های لازم را فراهم می نماییم که این نوع مدل برای رتبه بندی در آنها استفاده می شود . بر اساس الگوریتم Naïve Bayes ما می توانیم روش AdaBoost.M1 را برای همه معاملات بکار بندیم . عملیات زیر نوعی رتبه بندی برای مجموعه آزمایشات فراهم می کند . در حقیقت 69% از فراخوانی دارای 1.5 امتیاز خوب بر حسب میانگین NDTP است . منحنی های فراخوانی و PR به صورت زیر در نظر گرفته می شوند :


```

> par(mfrow=c(1,2))
> info <- attr(ab.res,'itsInfo')
> PTs.ab <- aperm(array(unlist(info),dim=c(length(info[[1]]),2,3)),
+                 c(1,3,2)
+                 )
> PRcurve(PTs.nb[, ,1],PTs.nb[, ,2],
+         main='PR curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> PRcurve(PTs.orh[, ,1],PTs.orh[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')

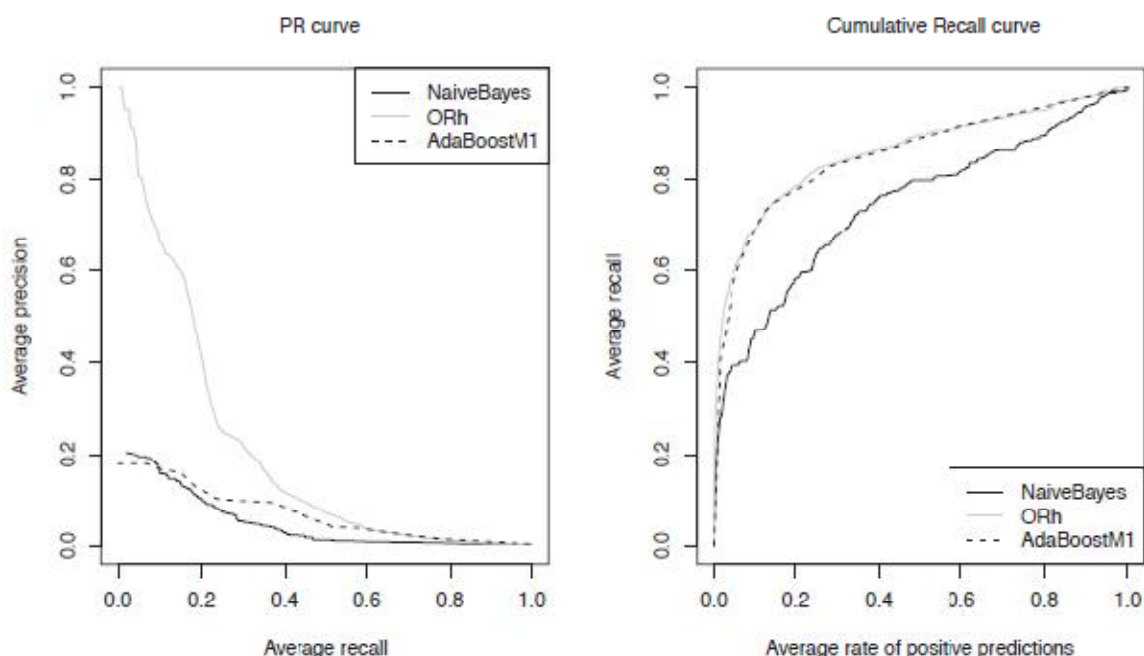
> PRcurve(PTs.ab[, ,1],PTs.ab[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> legend('topright',c('NaiveBayes','ORh','AdaBoostM1'),
+       lty=c(1,1,2),col=c('black','grey','black'))
> CRchart(PTs.nb[, ,1],PTs.nb[, ,2],
+         main='Cumulative Recall curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> CRchart(PTs.orh[, ,1],PTs.orh[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')
> CRchart(PTs.ab[, ,1],PTs.ab[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> legend('bottomright',c('NaiveBayes','ORh','AdaBoostM1'),
+       lty=c(1,1,2),col=c('black','grey','black'))

```

نتایج حاصله بهترین نتایج محسوب می شوند . در حقیقت این امتیازات با امتیازات دیگر مقایسه شده که بر اساس روش های LOF و ORh است . همچنین باید متذکر شد که این مدل تنها برای بخش های کوچک استفاده می شود تا رتبه بندی آنها حاصل گردد . علیرغم این مسئله ، در حدود 69% فراخوانی دارای 1.5 امتیاز خوب بوده که بر حسب میانگین NDTP است .

نمودارها در شکل 4-13 تأیید کننده عملکرد بالای الگوریتم AdaBoost.M1 است به ویژه بر حسب منحنی فراخوانی تراکمی که این مسئله بیشتر مشخص می شود . منحنی فوق نشان می دهد که برای بیشتر سطوح کار ، روش AdaBoost.M1 موجب ایجاد امتیاز حاصله توسط ORh می گردد . بر اساس وضعیت دقت / فراخوانی ، عملکرد AdaBoost.M1 قابل توجه نیست به ویژه برای سطوح پایین که فراخوانی وجود دارد . به هر حال

برای سطوح بالاتر فراخوانی ، موارد دقت بالاترین امتیاز را داشته و بهتر است حاصل گردد . اضافه بر آن ، باید توجه نمود که این مقادیر از سطوح بالا دقیقا" برای کاربرد ما لازم می باشد .



شکل 4-13: منحنی های فراخوانی (سمت راست) و PR (سمت چپ) برای دو بخش دیگر روش های ORh و Naïve Bayes و AdaBoost.M1

به طور خلاصه باید بیان کرد که روش AdaBoost.M1 روشی بسیار کاربردی محسوب می شود . علیرغم مشکلات عدم تعادل طبقه ، این روش برای دسترسی به عملکردهایی که دقت بالا لازم دارد مناسب است تا رتبه بندی هایی حاصل گردد .

4-4-3: رویکردهای نیمه نظارتی :

این بخش به شرح و بررسی استفاده از گزارشاتی می پردازد که بازبینی شده و یا بازبینی نشده اند و باید مدل طبقه بندی کسب کنند تا گزارشات دروغین و جعلی را شناسایی نمایند . روش های نیمه نظارتی از طریق مشاهدات ایجاد می شوند که کاربردهای متعددی در یافتن اطلاعات دارند ؛ نمونه هایی که دارای مقدار تعیین شده ای متغیر هدف می باشند . معمولا" این اطلاعات نیازمند فعالیت کارشناسان است و موجب افزایش هزینه های جمع آوری اطلاعات می گردد . از طرف دیگر ، کسب اطلاعات غیر مشخص آسان است به ویژه اینکه استفاده گسترده ای از سنسورها و دیگر انواع ابزارهای جمع آوری اطلاعات اتوماتیک می گردد . روش های نیمه نظارتی نیز در این بخش قابل استفاده هستند چرا که می توانند این نوع مجموعه اطلاعات را در نمونه های

مشخص و غیر مشخص ارائه دهند . معمولاً دو نوع روش نیمه نظارتی وجود دارد . از یک طرف روش های طبقه بندی نیمه نظارتی که تلاش بر بهبود عملکرد الگوریتم های طبقه بندی استاندارد نظارتی دارد و در کسب اطلاعات از نمونه های غیر مشخص کمک می نماید . رویکرد متناوب در واقع روش های طبقه بندی نیمه نظارتی بر اساس اطلاعات مشخص جهت معیارهایی لازم برای شکل گیری گروه ها است . در گروه بندی یا دسته بندی نیمه نظارتی ، ایده اصلی بر سر استفاده از اطلاعات در دسترس برای فرایند دسته بندی است که شامل نمونه هایی با همان اطلاعات و همان گروه می باشد و یا نمونه هایی با اطلاعات مختلف در گروه های مختلف است . معیار مورد استفاده جهت شکل گیری دسته ها ، تغییر روش ها و یافتن گروه های مناسب برای نمونه ها است . بر اساس رویکردهای نیمه نظارتی مشابه ، استفاده از الگوریتم ها می تواند موجب بهینه سازی محدودیت ها گردد . با توجه به طبقه بندی نیمه نظارتی ، روش شناسی های مختلفی وجود دارد . یکی از روش های شناخته شده ، خود آموزشی است . این رویکرد نوعی رویکرد تعاملی است . مرحله بعد استفاده از مدل برای طبقه بندی اطلاعات نامشخص است . نمونه هایی که مدل میزان اطمینان بیشتر دارد به اطلاعات نامشخص تقسیم می شوند . با استفاده از مجموعه جدید مدل جدیدی بدست می آید و مورد پردازش قرار می گیرد تا برخی معیارها حاصل گردند . از نمونه های مدل های طبقه بندی نیمه نظارتی می توان به TSVMs اشاره کرد . هدف از TSVMs کسب نمونه هایی برای مجموعه اطلاعات نامشخص است . مجدداً باید محدودیت های خاص کاربردها را در نظر بگیریم که بر حسب رتبه بندی می باشند . این امر موجب می شود تا از استراتژی های لازم برای روش های نظارتی و غیر نظارتی استفاده شود که این مسئله به طبقه بندی های اطلاعات مربوط می شود .

```

> ts <- iris[-idx, ]
> nb <- naiveBayes(Species ~ ., tr)
> table(predict(nb, ts), ts$Species)

```

	setosa	versicolor	virginica
setosa	12	0	0
versicolor	0	21	1
virginica	0	0	16

```

> trST <- tr
> nas <- sample(100, 90)
> trST[nas, "Species"] <- NA
> func <- function(m, d) {
+   p <- predict(m, d, type = "raw")
+   data.frame(cl = colnames(p)[apply(p, 1, which.max)],
+             p = apply(p, 1, max))
+ }
> nbSTbase <- naiveBayes(Species ~ ., trST[-nas, ])
> table(predict(nbSTbase, ts), ts$Species)

```

	setosa	versicolor	virginica
setosa	12	0	0
versicolor	0	18	2
virginica	0	3	15

```

> nbST <- SelfTrain(Species ~ ., trST, learner("naiveBayes",
+   list()), "func")
> table(predict(nbST, ts), ts$Species)

```

	setosa	versicolor	virginica
setosa	12	0	0
versicolor	0	20	2
virginica	0	1	15

```

> library(DMwR)
> library(e1071)
> data(iris)
> idx <- sample(150, 100)
> tr <- iris[idx, ]

```

کد بالا شامل مراحل مختلف از مدل Naïve Bayes است و 100 نمونه را مورد بررسی قرار می دهد. همانطور که در بخش 3-3-4 مشاهده نمودید ، هنگامی که هدف ما پیش بینی مجموعه کوچک از رخدادهای کمیاب

است ، دقت و فراخوانی نوعی اندازه گیری در ارزیابی مناسب محسوب می شود . با توجه به محدودیت بازبینی k ، ما می توانیم دقت و بازخوانی بالاترین موقعیت k را در رتبه بندی محاسبه نماییم . این مقدار یا حد k تعیین کننده این موضوع است که کدام گزارشات بر اساس مدلسازی مورد بازبینی قرار گرفته اند . با توجه به دیدگاه طبقه بندی شده موقعیت k در موارد جعلی قابل شناسایی است به طوریکه باقیمانده گزارشات طبیعی هستند . مقدار دقت به ما می گوید که چه میزان گزارش موقعیت k جعلی است . مقدار فراخوانی به ارزیابی تعداد گزارشات جعلی در مجموعه آزمایشات می پردازد . باید به این نکته توجه نمود که مقادیر حاصله نوعی بدبینی را نشان می دهد . در حقیقت اگر موقعیت k شامل گزارشات غیر مشخص باشد ما نمی توانیم دقت و فراخوانی را محاسبه نماییم . به هر حال ، اگر گزارشات بازنگری شوند می توانیم مقادیر واقعی را پیدا نماییم .

```

> pred.nb <- function(m,d) {
+   p <- predict(m,d,type='raw')
+   data.frame(cl=colnames(p)[apply(p,1,which.max)],
+             p=apply(p,1,max)
+             )
+ }
> nb.st <- function(train,test) {
+   require(e1071,quietly=T)
+   train <- train[,c('ID','Prod','Uprice','Insp')]
+   train[which(train$Insp == 'unkn'),'Insp'] <- NA
+   train$Insp <- factor(train$Insp,levels=c('ok','fraud'))
+   model <- SelfTrain(Insp ~ .,train,
+                     learner('naiveBayes',list()),'pred.nb')
+   preds <- predict(model,test[,c('ID','Prod','Uprice','Insp')],
+                   type='raw')
+   return(list(rankOrder=order(preds['fraud'],decreasing=T),
+             rankScore=preds['fraud'])
+           )
+ }
> ho.nb.st <- function(form, train, test, ...) {
+   res <- nb.st(train,test)
+   structure(evalOutlierRanking(test,res$rankOrder,...),
+             itsInfo=list(preds=res$rankScore,
+                          trues=ifelse(test$Insp=='fraud',1,0)
+             )
+           )
+ }
> nb.st.res <- holdOut(learner('ho.nb.st',
+                             pars=list(Threshold=0.1,
+                                       statsProds=globalStats)),
+                   dataset(Insp ~ .,sales),
+                   hldSettings(3,0.3,1234,T),
+                   itsInfo=TRUE
+                   )

```

نتایج مدل self trained به صورت زیر می باشد :

```

> summary(nb.st.res)

== Summary of a Hold Out Experiment ==

Stratified 3 x 70 %/ 30 % Holdout run with seed = 1234

* Dataset :: sales
* Learner  :: ho.nb.st with parameters:
    Threshold = 0.1
    statsProds = 11.34 ...

* Summary of Experiment Results:

      Precision      Recall      avgNDTP
avg      0.013521017 0.42513271 1.08220611
std      0.001346477 0.03895915 1.59726790
min      0.012077295 0.38666667 0.06717087
max      0.014742629 0.46456693 2.92334375
invalid 0.000000000 0.00000000 0.00000000

```

به هر حال ، چنین استراتژی به ضرورت منجر به دقت پایین می شود . کاربرد فعلی دارای جزئیات متعددی است . با توجه به این حقیقت که محدودیت هایی در منابع سرمایه گذاری شده در فعالیت های بازاریابی وجود دارد ، آنچه که ما واقعا "خواستار آن هستیم افزایش استفاده از منابع است . این بدان معناست که اگر ما X ساعت جهت بازاریابی گزارشات صرف نماییم و اگر X ساعت گزارش طبیعی بازاریابی شود آنگاه در رتبه بندی ما دقت پایین است . فراخوانی دقیقا "موضوع اصلی در این بخش است و آنچه که ما قادریم تا کسب نماییم در واقع 100% منابع در دسترس است .

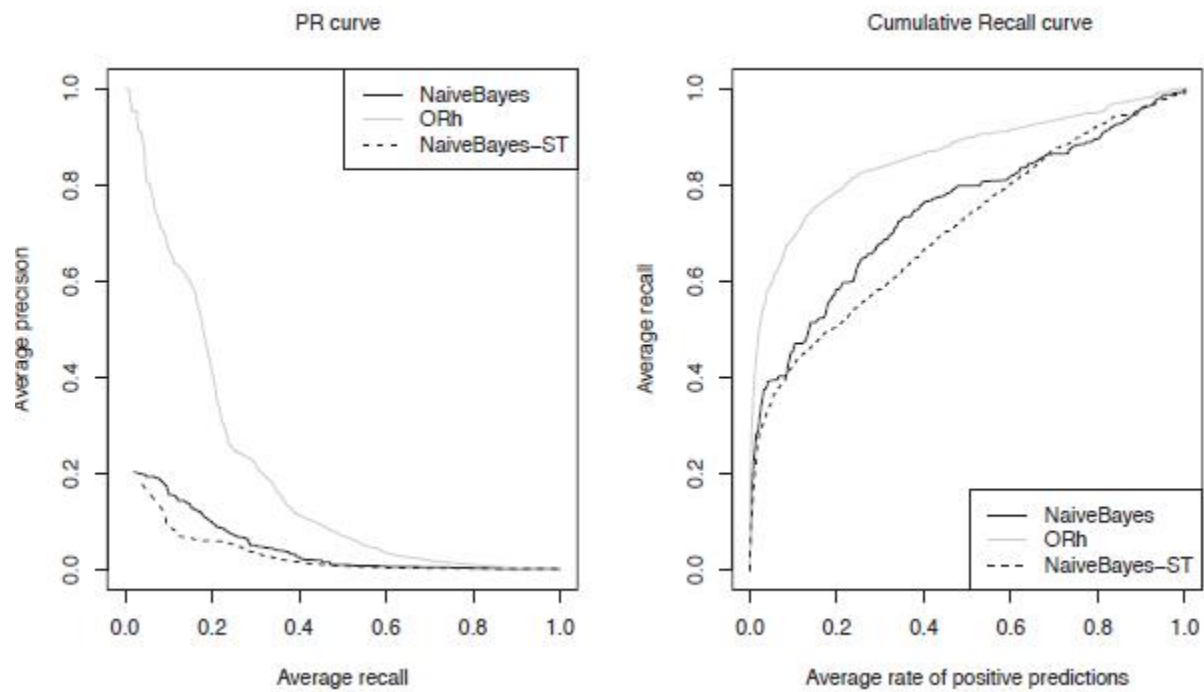
نتایج حاصله ممکن است رضایت بخش نباشد . تشابهات زیادی با نتایج حاصله در مدل Naïve Bayes دارد که تنها از طریق اطلاعات مشخص قابل انجام است . با توجه به میانگین NDTP که وضعیت بهبود شده ای دارد ، همه آمارها با هم متشابه هستند و بر این اساس بهترین نتایج حاصل شده است .

```

+         add=T,lty=1,col='grey',
+         avg='vertical')
> PRcurve(PTs.nb.st[, ,1],PTs.nb.st[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> legend('topright',c('NaiveBayes','ORh','NaiveBayes-ST'),
+         lty=c(1,1,2),col=c('black','grey','black'))
> CRchart(PTs.nb[, ,1],PTs.nb[, ,2],
+         main='Cumulative Recall curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> CRchart(PTs.orh[, ,1],PTs.orh[, ,2],
+         add=T,lty=1,col='grey',
+         avg='vertical')
> CRchart(PTs.nb.st[, ,1],PTs.nb.st[, ,2],
+         add=T,lty=2,
+         avg='vertical')
> legend('bottomright',c('NaiveBayes','ORh','NaiveBayes-ST'),
+         lty=c(1,1,2),col=c('black','grey','black'))
> par(mfrow=c(1,2))
> info <- attr(nb.st.res,'itsInfo')
> PTs.nb.st <- aperm(array(unlist(info),dim=c(length(info[[1]]),2,3)),
+                     c(1,3,2)
+                     )
> PRcurve(PTs.nb[, ,1],PTs.nb[, ,2],
+         main='PR curve',lty=1,xlim=c(0,1),ylim=c(0,1),
+         avg='vertical')
> PRcurve(PTs.orh[, ,1],PTs.orh[, ,2],

```

نمودارها ، تائید کننده عملکرد ناامید کننده طبقه بندی کننده self trained Naïve Bayes است . برای این وضعیت خاص ، طبقه بندی کننده نیمه نظارتی به طور شفاف نمی تواند با مدل Naïve Bayes استاندارد مقایسه شود که بر اساس مجموعه اطلاعات کوچکتر می باشد .



شکل 4-14: منحنی های فراخوانی (سمت راست) و PR (سمت چپ) برای روش های ORh و self
 strained Naïve BAYES و روش های Naïve Bayes استاندارد و ORh
 ما همچنین از رویکرد self training با الگوریتم AdaBoost.M1 استفاده می نماییم . کد زیر آزمایشات زیر
 را نشان می دهد :


```

> pred.ada <- function(m,d) {
+   p <- predict(m,d,type='probability')
+   data.frame(cl=colnames(p)[apply(p,1,which.max)],
+             p=apply(p,1,max)
+           )
+ }
> ab.st <- function(train,test) {
+   require(RWeka,quietly=T)
+   train <- train[,c('ID','Prod','Uprice','Insp')]
+   train[which(train$Insp == 'unkn'),'Insp'] <- NA
+   train$Insp <- factor(train$Insp,levels=c('ok','fraud'))
+   model <- SelfTrain(Insp ~ .,train,
+                     learner('AdaBoostM1',
+                           list(control=Weka_control(I=100))),
+                     'pred.ada')
+   preds <- predict(model,test[,c('ID','Prod','Uprice','Insp')],
+                   type='probability')
+   return(list(rankOrder=order(preds['fraud'],decreasing=T),
+             rankScore=preds['fraud'])
+         )
+ }
> ho.ab.st <- function(form, train, test, ...) {
+   res <- ab.st(train,test)
+   structure(evalOutlierRanking(test,res$rankOrder,...),
+             itInfo=list(preds=res$rankScore,
+                       trues=ifelse(test$Insp=='fraud',1,0)
+             )
+         )
+ }
> ab.st.res <- holdOut(learner('ho.ab.st',
+                             pars=list(Threshold=0.1,
+                                       statsProds=globalStats)),
+                   dataset(Insp ~ .,sales),
+                   hldSettings(3,0.3,1234,T),
+                   itsInfo=TRUE
+                 )

```

با استفاده از این رویکرد و الگوریتم AdaBoost.M1 نتایج مطلوب زیر حاصل شده است . این نتایج برای 10% تلاش در نظر گرفته شده است .

نتایج self trained AdaBoost برای 10% تلاش به صورت زیر می باشد :

```

> summary(ab.st.res)

== Summary of a Hold Out Experiment ==

Stratified 3 x 70 %/ 30 % Holdout run with seed = 1234

* Dataset :: sales
* Learner :: ho.ab.st with parameters:
  Threshold = 0.1
  statsProds = 11.34 ...

* Summary of Experiment Results:

      Precision    Recall  avgNDTP
avg      0.022377700 0.70365350 1.6552619
std      0.001130846 0.02255686 1.5556444
min      0.021322672 0.68266667 0.5070082
max      0.023571548 0.72750643 3.4257016
invalid 0.000000000 0.00000000 0.0000000

```

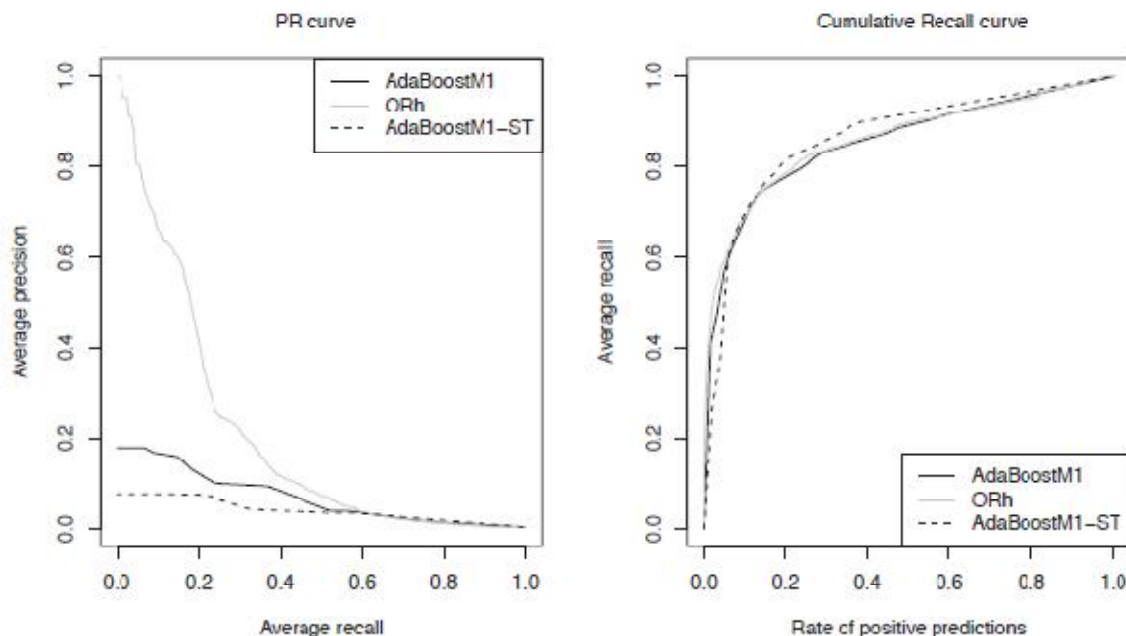
با بررسی نتایج مذکور باید گفت که گرچه این امتیازات موثر نیستند اما نشان دهنده بهبودی مناسب بر اساس AdaBoost.M1 هستند که با استفاده از اطلاعات مشخص حاصل شده است . در این بخش دقت به همان میزان باقی می ماند و بهبودی کم در بخش فراخوان و NDTP است . مقدار فراخوان بالاتر بوده و بر اساس 10% تلاش است .

```

> par(mfrow = c(1, 2))
> info <- attr(ab.st.res, "itsInfo")
> PTs.ab.st <- aperm(array(unlist(info), dim = c(length(info[[1]]),
+   2, 3)), c(1, 3, 2))
> PRcurve(PTs.ab[, , 1], PTs.ab[, , 2], main = "PR curve",
+   lty = 1, xlim = c(0, 1), ylim = c(0, 1), avg = "vertical")
> PRcurve(PTs.orh[, , 1], PTs.orh[, , 2], add = T, lty = 1,
+   col = "grey", avg = "vertical")
> PRcurve(PTs.ab.st[, , 1], PTs.ab.st[, , 2], add = T, lty = 2,
+   avg = "vertical")
> legend("topright", c("AdaBoostM1", "ORh", "AdaBoostM1-ST"),
+   lty = c(1, 1, 2), col = c("black", "grey", "black"))
> CRchart(PTs.ab[, , 1], PTs.ab[, , 2], main = "Cumulative Recall curve",
+   lty = 1, xlim = c(0, 1), ylim = c(0, 1), avg = "vertical")
> CRchart(PTs.orh[, , 1], PTs.orh[, , 2], add = T, lty = 1,
+   col = "grey", avg = "vertical")
> CRchart(PTs.ab.st[, , 1], PTs.ab.st[, , 2], add = T, lty = 2,
+   avg = "vertical")
> legend("bottomright", c("AdaBoostM1", "ORh", "AdaBoostM1-ST"),
+   lty = c(1, 1, 2), col = c("black", "grey", "black"))

```

شکل 15-4 ، منحنی های مدل های مذکور است که با روش های استاندارد ORh و AdaBoost.M1 همخوانی دارد . منحنی های حاصله طبیعی هستند . منحنی ذکر شده تأیید می نماید که AdaBoost.M1 بهترین مدل یا روش برای شناسایی گزارشات جعلی است . برای بازبینی از طریق این روش تنها 15 تا 20 درصد بیشتر از دیگر سیستم ها موارد جعلی تشخیص داده می شود . بر اساس دقت ، امتیازات جالب نمی باشند اما اگر گزارشات نامشخص در آن قرار بگیرد رتبه بندی ها در مورد موارد دروغین تأیید کننده است .



شکل 4-15: منحنی های فراخوانی (سمت راست) و PR (سمت چپ) از AdaBoost.M1 باروش های ORh و AdaBoost.m1 استاندارد

4-5: خلاصه مطالب :

هدف اصلی این مقاله ، کمک به خواننده جهت ایجاد طبقه ای جدید برای استخراج اطلاعات غلط می باشد که به آن رتبه بندی خطا گفته می شود . شناسایی تقلب و فریبکاری در واقع بخش مهمی برای کاربرد بالقوه تکنیک های استخراج اطلاعات می باشد که نتایج اقتصادی و اجتماعی دارد و معمولاً با فعالیت های غیر قانونی همراه است . بر اساس دیدگاه تحلیل اطلاعات ، فعالیت های متقلبانه معمولاً با مشاهدات غیر معمول همراه است و فعالیت هایی هستند که از حالت نرمال و هنجار دور شده و منحرف گشته اند . چنین انحرافات که از مسیر طبیعی خارج گشته اند به صورت کلی امور غیر متعارفی در اصول مختلف تحلیل اطلاعات محسوب می شوند . در حقیقت ، تعریف استاندارد از امور غیر متعارف این است : مشاهداتی که نسبت به دیگر موارد مشابه از مسیر نرمال منحرف گشته اند و موجب برانگیختگی بدگمانی ها می گردد و بوسیله مکانیسم مختلفی ارتقا می یابد . ما قصد داریم نوعی رتبه بندی محتمل بر تقلب را به عنوان پیامد فرایند فراهم نماییم . این رتبه بندی ها این امکان را به شرکت می دهند تا از طریق روش های سودمند منابع خود را بازرسی کنند . چنین فعالیت هایی در حوزه های مختلف و به طور متناوب قابل انجام است . چهارچوب اصلی کار نیاز دارد تا به طراحی نقشه پرداخته شود تا بر اساس آن موارد کاربردی مشخص گردند و در این راه از بخش های زیر استفاده می شود : شناسایی خطا در معاملات ، شناسایی خطا در ارتباطات ، شناسایی خطا در پرداخت مالیات و در حوزه ایمنی نیز این مفاهیم کاربردهای متعددی دارند . بر اساس روش شناسی ، ما موضوعات زیر را ارائه داده ایم :

- شناسایی و رتبه بندی خطا

- روش های دسته بندی
- یادگیری نیمه نظارتی
- طبقه بندی نیمه نظارتی بر اساس خود آموزشی
- توزیع طبقه غیر متعادل و روش های لازم برای ارائه انواع مشکلات
- طبقه بندی کننده Naïve Bayes
- طبقه بندی کننده AdaBoost
- فراخوان / دقت و منحنی های تراکمی
- آزمایشات hold out

بر اساس دیدگاه یادگیری R ما به مطالب زیر اشاره نمودیم :

- چگونه به ارزیابی مختلف آماری پردازیم و چگونه با استفاده از بسته ROCR موضوعات را تجسم نماییم؟
- چگونه تخمین های hold out را برای متریک های تخمین بدست آوریم ؟
- چگونه با استفاده از روش LOF بتوانیم فاکتورهای داخلی خطا را بیابیم؟
- چگونه با استفاده از روش ORh بتوانیم خطاها را رتبه بندی کنیم ؟
- چگونه با روش SMOTE بتوانیم عدم تعادل را پیدا کنیم ؟
- چگونه با استفاده از مدل های Naïve Bayes به طبقه بندی دست پیدا کنیم ؟
- چگونه طبقه بندی کننده های AdaBoost.M1 را بدست آوریم ؟
- چگونه از طبقه بندی مجموعه اطلاعات نیمه نظارتی برای خود یادگیری استفاده نماییم ؟

دو جمله اول از نام ها و توضیحات co variate ها بدست می آیند . سپس برخی اطلاعات بر اساس توزیع نمونه های دو co variate اصلی حاصل می گردد : متغیر BT که تعیین کننده نوع lymphoblastic leukemia اصلی و متغیر mol.bio است که شرح غیر طبیعی بودن سیتوژنیک را ارائه می دهد و در هر نمونه یافت می شود .

کد :

کد بالا نشان دهنده نام های 10 زن و نام های 5 نمونه است .

همانطور که قبلاً ذکر شد ما در تحقیق خود روی نمونه های B- cell ALL تاکید داشته و به طور خاص روی نمونه های زیر مجموعه اشاره دارد که نمونه های هدف محسوب می شوند . کد زیر شامل مجموعه اطلاعات مورد استفاده است .

کد:

جمله اول ارائه دهنده مجموعه نمونه های در نظر گرفته شده است . این نمونه ها دارای مقادیر خاص متغیرهای BT و mol.bio هستند . با استفاده از عملیات table() تمامی نمونه ها را بررسی کنید . سپس زیر مجموعه را برای بخش ALL در نظر بگیرید تا 94 نمونه حاصل گردد . زیر مجموعه مذکور شامل برخی مقادیر متغیرهای BT و mol.bio است . در این بخش ما باید سطوح دردسترس را بر اساس دو فاکتور در ALLb به روز نماییم .

کد :

ALLb مجموعه اطلاعات است که در سرتاسر این متن به آن اشاره شده است . ممکن است که ذخیره کردن این بخش در فایل محلی کامپیوتر شما ایده خوبی باشد و شما نیازی به تکرار مراحل پیش پردازش نداشته باشید و تحلیل را با استفاده از برنامه زیر شروع نمایید.

کد :

5-2-1 : کشف مجموعه اطلاعات :

عملیات exprs() این امکان را به ما می دهد تا ماتریکس سطوح بیان کننده ژن را بدست آوریم .

کد :

IranDataMiner.ir

